NIAKWA PROGRAMMING LANGUAGE

REVISION 4.10 ADDENDUM

FOR MS-DOS PLATFORMS

1st Edition - March 1994

Niakwa, Inc.
23600 N. Milwaukee Avenue
Mundelein, IL 60060

PHONE (708) 634-8700   FAX (708) 634-8718   TELEX 3719965 NIAK UB

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES AND PROPRIETARY RIGHTS

The staff of Niakwa, Inc. (Niakwa) has taken due care in preparing this manual. Nothing contained herein shall be construed to modify or alter in any way the standard terms and conditions of the Niakwa Programming Language (NPL) Support and Distribution License Agreement, the End-User Support Only License Agreement, the Niakwa Software License Agreement and Warranty and any other Niakwa License Agreement (collectively, the "License Agreements") by which this software package was acquired.

This manual is to serve as a guide for use of the Niakwa software only and not as a source of representations or additional undertakings by Niakwa. The licensee must refer to the License Agreements for Niakwa product and service representations.

No ownership of Niakwa software is transferred by any of the License Agreements. Any use of Niakwa software beyond the terms and conditions of the License Agreements, without the written authorization of Niakwa, is prohibited.

PREFACE

The Revision 4.10 Addendum for the Niakwa Programming Language (NPL) is designed as an addition to the NPL MS-DOS Supplement. This addendum discusses the features and enhancements that have been added to NPL since the original NPL 4.00 release for MS-DOS based platforms. Refer to the appropriate NPL platform-specific addendum for operating environment-specific features.

The MS-DOS environments currently supported by NPL Revision 4.10 include:

Hardware	**Operating Systems**	**Operating Environments**
IBM-Compatible PC	MS-DOS 2.0 or greater	MS-DOS
IBM-Compatible PC	MS-DOS 2.0 or greater	Novell NetWare 2.10 or greater
IBM-Compatible PC (80286 or greater)	MS-DOS 3.1 or greater	MS-Windows 3.1 or greater
IBM-Compatible PC (80386 or greater)	MS-DOS 3.1 or greater	Phar Lap 386/DOS Extender
IBM-Compatible PC (80286 or greater)	MS-DOS 3.1 or greater	NetBIOS compatible environment (i.e., NetWare Lite, LANtastic, etc.)

TABLE OF CONTENTS

PREFACE

NPL ENHANCEMENTS

CHAPTER 1

NPL ENHANCEMENTS

1.1 Overview

Revision 4.10 of the Niakwa Programming Language contains a number of significant enhancements over the original Revision 4.00 version. This addendum documents all new product enhancements implemented with Revision 4.10.

Section 1.2 discusses the contents of the NPL 4.10 Update Package.

Section 1.3 discusses the installation of the NPL 4.10 Update Pacakage.

Section 1.4 discusses NPL 4.10 bug corrections.

Section 1.5 discusses NPL 4.10 security issues.

Section 1.6 discusses NPL 4.10 new and enhanced features.

Section 1.7 discussses NPL 4.10 new and update system variables.

1.2 Contents

Revision 4.10 of the Niakwa Programming Language required the modification of a number of NPL files. However, these files continue to have the same filenames as documented in Chapter 1 of the NPL Release IV MS-DOS Supplement. In addition to these files, several new files have been added with Revision 4.10 as shown below.

1.2.1 Revision 4.10 Development Package Contents

All files remain as documented in Chapter 1 of the NPL Release IV MS-DOS Supplement. No new development files have been added to this release of NPL.

1.2.2 Revision 4.10 RunTime Package Contents

All files remain as documented in Chapter 1 of the NPL Release IV MS-DOS Supplement, plus the following:

Master Gold Key (Disk 1 of 2)

NIAKNETB.COM	TSR program used to enable NETBIOS support. Refer to the NPL NetBIOS Addendum for details.

MS-Windows RunTime Diskette

RTISLAVE.EXE	This file is used by RTIWIN.EXE to invoke the new debug screen under MS-Windows. Refer to Section 1.6 for a detailed discussion of this new feature.

1.3 Installation

The changes associated with NPL Revision 4.10 affect both the NPL Development and RunTime Packages. However, the loading instructions for the NPL Revision 4.10 Development and RunTime Packages remain the same as those documented in Chapter 2 of the NPL Release IV MS-DOS Supplement.

NOTE: Revision 4.10 introduces a number of operational differences in how the RunTime handles file sharing and maintains the current user count. Refer to Section 1.5 for details.

1.4 Bug Corrections

All corrected bugs, noted as corrected in the NPL Bug Reports, have been applied to NPL Revision 4.10.

NOTE: The NPL Bug Reports are updated monthly on the Niakwa BBS and are distributed to the field quarterly. In addition to the NPL Bug Reports, an index of all reported bugs is maintained on the Niakwa BBS. This provides a short description of the bug, the number of the Bug Report in which it was originally published, and the internal NPL revision in which it was corrected. For details on accessing the Niakwa BBS, please contact Niakwa Technical Support.

1.5 NPL Security

NPL Revision 4.10 has been updated to provide support for the growing number of NetBIOS-compatible environments. In addition, the startup functionality of the RunTime has be modified, by using the NPL security check mechanism, for file sharing and user count determination. These enhancements allow the RunTime an increased flexibility in adapting to the complex multi-user configurations that are becoming common.

NOTE: Although the NPL security check is now used by the RunTime to determine some of its startup functionality, the installation process of NPL Gold Key security has not changed from previous revisions (refer to Chapter 2 of the NPL Release IV MS-DOS Supplement).

1.5.1 User Count and File Sharing Determination

To understand how the RunTime determines the mode in which it opens a file and whether or not the user count is incremented, it is necessary to understand the components that make up the decision tree logic NPL uses at startup.

Two Types of RunTimes

Previous versions of NPL for MS-DOS environments were distributed as single-user MS-DOS RunTimes or multi-user Novell NetWare RunTimes. As of NPL Revision 4.10, NPL for MS-DOS environments is distributed as a NetBIOS RunTime (available in 1 to 8-user increments) or a Niakwa Networks RunTime. The two types of RunTimes are described below:

NetBIOS RunTime — Operates in a single user MS-DOS or multi-user NetBIOS environment.

Niakwa Networks RunTime — Has all the capabilities of the NetBIOS RunTime plus the ability to operate on a Novell NetWare network.

NOTE: All single-user RunTimes are treated as Niakwa Networks RunTimes. This conforms to the previous functionality of NPL MS-DOS RunTimes. The multi-user NetBIOS RunTimes (3, 4, and 8 user) will not operate when Novell NetWare drivers are detected and the NPL security is not passed locally (i.e., from the local hard drive or from the Gold Key diskette).

Three File Opening Modes

The NPL Revision 4.10 RunTime for MS-DOS environments is able to open files in three different modes as shown below:

EXCLUSIVE — This is the mode in which all single-user NPL RunTimes would open files prior to Revision 4.10 of NPL. In this mode, only one workstation can access the file (i.e., other tasks cause an error if they attempt to access open files).

SHARE — NPL accesses files using MS-DOS SHARE-compatible lock calls.

NOVELL — NPL opens files in shared mode and accesses files using Novell NetWare-compatible lock calls.

NOTE: NPL Revision 4.10 always opens local files in shared mode and accesses these files using MS-DOS SHARE-compatible lock calls.

User Counts

NPL Revision 4.10 for MS-DOS environments provides additional methods for tracking the number of active NPL 4.10 users on the system. These methods are described below:

FINGERPRINT

The user count is not tracked. This mode is invoked when the security is passed locally (i.e., from a local drive or Gold Key diskette) and the NIAKNETB TSR is not present. Refer to the NPL NetBIOS Addendum for details on the NIAKNETB TSR.

For example, a laptop with single-user NPL RunTime installed on it could be connected to a network with a 4-user NPL RunTime installed on it. Execution of the single user RunTime would not increment the user count on the network.

HINT: This method can also be used with NetBIOS environments that may encounter conflicts with the NIAKNETB TSR. In this scenario, a single-user NPL RunTime could be installed at each workstation and security would be passed locally. In this case, file sharing and other multi-user features are enabled.

NIAKNETB

The user count is incremented by NIAKNETB. This mode is invoked when the NIAKNETB TSR is present and Novell NetWare drivers are not detected. Refer to the NPL NetBIOS Addendum for details.

SEMAPHORE

The user count is incremented by a NetWare Semaphore. This mode is invoked upon detection of a Niakwa Networks RunTime (refer above); Novell NetWare drivers are present, and security is not passed locally (i.e., from a local drive or Gold Key diskette).

UNCOUNTED

The user count is not tracked. This mode is invoked only on a Niakwa Networks RunTime (refer above) when no network is detected (either Novell or NIAKNETB) and security is passed from a non-local drive. In this case, file sharing and other multi-user features are not enabled. Here, all files are opened in exclusive mode.

WARNING -- Some environments open files in shared mode even if exclusive mode is requested. In this type of environment, multiple NPL users could access files concurrently. However, since NPL file locking calls are not issued in these environments (NPL thinks the file is open in exclusive mode), data corruption could occur.

As the above methods illustrate, NPL Revision 4.10 for MS-DOS environments can conform to many different configurations. To simplify system planning, the following tables have been designed to provide developers with a clear layout of the startup functionality of NPL.

Table 1 illustrates the startup behavior of a NetBIOS RunTime. The first three columns represent the environment considerations NPL takes into account at startup, while the last two columns illustrate the user count and file sharing behavior (for non-local files) used by the RunTime. Table 2 illustrates the startup behavior of a Niakwa Networks RunTime.

NetBIOS RunTime

RunTime Environment			**Startup Behavior**	
NetWare Detected	**NIAKNETB Detected ***	**Fingerprint Local**	**Count Method**	**Share Method**
N	N	N	WON'T RUN - 0	N/A
N	N	Y	FINGERPRINT (LOCAL)	SHARE
N	Y	N	NIAKNETB	SHARE
N	Y	Y	NIAKNETB	SHARE
Y	N	N	WON'T RUN - 1	N/A
Y	N	Y	FINGERPRINT (LOCAL)	Novell
Y	Y	N	WON'T RUN - 1	N/A
Y	Y	Y	FINGERPRINT (LOCAL)	Novell

Table 1

* The presence of NIAKNETB.COM is ignored if the user limit is 1.

Niakwa Networks RunTime

RunTime Environment			**Startup Behavior**	
NetWare Detected	**NIAKNETB Detected ***	**Fingerprint Local**	**Count Method**	**Share Method**
N	N	N	UNCOUNTED	EXCLUSIVE
N	N	Y	FINGERPRINT (LOCAL)	SHARE
N	Y	N	NIAKNETB	SHARE
N	Y	Y	NIAKNETB	SHARE
Y	N	N	SEMAPHORE	Novell
Y	N	Y	FINGERPRINT (LOCAL)	Novell
Y	Y	N	SEMAPHORE	Novell
Y	Y	Y	FINGERPRINT (LOCAL)	Novell

Table 2

* The presence of NIAKNETB.COM is ignored if the user limit is 1.

Description of Count Method

FINGERPRINT (LOCAL) Refer to the description above under User Counts.

NIAKNETB Refer to the description above under User Counts.

SEMAPHORE Refer to the description above under User Counts.

WON’T RUN - 0 If user limit is not 1 and the NetBIOS RunTime is detected, exits with message:

```
NetBIOS is running but NIAKNETB.COM is not.
Authorization must be installed on a local drive.
```

If the user limit is 1, NPL runs with a Count Method of UNCOUNTED and a Share Method of EXCLUSIVE.

WON’T RUN - 1	Exits with message:

```
This NPL RunTime is not authorized for use with
Netware.
Authorization must be installed on a local drive.
```

In addition to using the tables above, developers may determine the status of the NPL RunTime environment and startup behavior under program control by using Byte 32 and 33 of $MACHINE. Byte 32 is used to examine the specific NPL RunTime environment encountered at startup, while Byte 33 is used to determine the specific startup behavior applied to the NPL RunTime at startup. Refer to Section 1.7 details.

1.5.2 Operational Differences from Previous Revisions

This section discusses several operational differences between NPL Revision 4.10 and previous versions. These operational changes have been made to enhance NPL’s support of multi-tasking and mixed network environments.

Shared versus Exclusive Access

As noted above, NPL opens files using the SHARE category of file locks for files on non-local drives (except where Novell NetWare is detected) when the NPL security is passed locally (i.e., from the local drive or Gold Key diskette). This is a significant change from previous NPL RunTime revisions which would always open files using the Exclusive category on single-user systems. This means that multiple workstations with single-user RunTimes installed locally may now share files with other RunTimes in mixed network environments.

NOTE: If the MS-DOS SHARE.EXE program is loaded, local files are now always opened in shared mode and NPL performs file locking. This means that NPL Revision 4.10 now supports safe, simultaneous access to local files by more than one DOS, Windows, or 386/DOS-Extender task (i.e., task-switches under MS-DOS, multiple DOS tasks under MS-Windows, DESQview, Novell DOS 7, etc.).

Implied $BREAK after Disk I/O

$OPTIONS byte 14 (implied $BREAK after disk I/O) is no longer ignored under MS-DOS. In a multi-session environments such as DOS tasks under MS-Windows or mixed network environments, setting this byte to HEX(01) improves interleaved access to files. The default remains HEX(00).

Unique Terminal Identification

If the NIAKNETB TSR is detected at the startup of the RunTime, $NETID is set to the Network Interface Card (NIC) adapter address (if one is present), enabling developers to establish unique terminal and partition values. Refer to Section 5.3 of the NPL Novell NetWare Addendum for details.

NOTE: In cases where NIAKNETB is not detected, unique terminal and partition values can be established using the BASIC2C_ID environment variable. Refer to Section 5.3 of the NPL Novell NetWare Addendum for details.

Previous revisions of NPL defaulted all non-Novell NetWare systems to a #ID value of 0. Refer to Section 5.3 of the NPL Release IV Novell NetWare Addendum for a complete discussion on establishing unique terminal identification.

1.6 Revision 4.10 New and Enhanced Features

This section discusses a variety of new and enhanced features implemented in NPL Revision 4.10.

1.6.1 MS-Windows Debug Mode

As of Revision 4.10, the MS-Windows Interpretive RunTime (RTIWIN.EXE) supports a dual window debug mode, which may be selected from choices on the RTIWIN Options Menu. When the Debug Window is active, all immediate mode operations update a separate text-only window (24 lines of 80 characters).

HINT: The Debug Window can be located and sized separately from the application display.

The Debug Window permits STEPping through applications that update the application display window without adversely affecting the contents of this window. Specifically, the following output always appears in the Debug Window when this option is selected:

- Output from LIST and TRACE statements directed to the screen.
- Output from STOP and END statements.
- The Immediate mode command prompt and command line entries.

- Unrecoverable error statement displays.
- Output from PRINT statements executed from immediate mode.

In addition, the Debug Window and the application have separate windows event queues. This means that the keyboard and mouse input for the two windows is effectively independent (i.e., the window with the input focus receives keys entered at that time).

Whenever an active program HALTs or executes a STOP or END statement, the Debug Window becomes the active window. Immediate mode commands must be entered into this window and output from LIST or PRINT statements, which would normally appear on the application window, are displayed on the Debug Window instead. TRACE output normally directed to the screen is always directed to the Debug Window.

NOTE: INPUT SCREEN statements executed by programs (not in Immediate Mode) always get information from the application window, not the Debug Window.

HINT: The Debug Window option is convenient for debugging many conventional NPL applications, especially when debugging routines which update the application window.

The Debug Window option is essential for debugging NPL applications which are event-driven. As described above, the Debug Window uses a separate application message queue aside from the NPL application. This allows the programmer to STOP within message handling routines, without disrupting the sequence of events in the application. The Debug Window remains visible even if the NPL application window has been hidden.

For proper operation, the Debug Window options requires that the RTISLAVE.EXE module be located in the same directory as RTIWIN.EXE.

NOTE: Byte 47 of $OPTIONS has been updated to allow for disabling the "Debug" option from the applications options menu. Refer to Section 1.7 for details.

Options of the Debug Window

The following options are available from the Debug Window.

HALT Equivalent to pressing CTRL-BREAK. This option tells NPL to halt execution at the first opportunity--usually at the start of the next NPL statement.

STEP	Available while HALTed. This command does the equivalent of a STEP# command, if STEP mode is not yet active. It then does the equivalent of an EXECUTE (single step) key.

NOTE: Any partially entered command is discarded.

CONTINUE	Available while HALTed. This does the equivalent of a CONTINUE command.

NOTE: Any partially entered command is discarded.

Hard Mode

When this option is selected (from the Debug Window's Options menu), any time RTIWIN waits for a key in the Debug Window, all other activity in MS-Windows is effectively halted.

NOTE: This is not usually required for conventional NPL applications.

For NPL applications which are event-driven, if the 'DisablePoll function has been called to disable implied polling of the message queue, Hard Mode takes effect in the Debug Window, even if it is not selected from the menu.

While an application is halted in Hard Mode, messages cannot be sent to other applications. This means that if a Debug Window is moved or resized, exposed areas of the screen may not be repainted until execution is resumed and the message queue is read.

In addition, if Hard Mode is active and the mouse cursor is moved outside the Debug Window, the mouse pointer's display form remains "stuck" in the shape typically used to indicate a resize operation.

1.6.2 New RTIWIN.INI Parameters

As of Revision 4.10 of the NPL MS-Windows RunTime, three new parameters have been added to the list of parameters available for the RTINWIN.INI file. The three new parameters are for use with the new Debug Window option of the interpretive version (RTIWIN.EXE) of the NPL MS-Windows RunTime. For a complete list of all RTIWIN.INI parameters, refer to Section 3.4 of the NPL MS-Windows Addendum.

DebugAutoSize

Purpose:

For the Debug Window, allows specification of the status of the autosize flag (on or off) and the font sizes to be used if off. In addition, the maximum width of the screen used by the application is specified (usually 80 columns, but may also be 132 for applications that select a wide display).

The **DebugAutoSize** option and font size may be changed by the user by resizing the windows. The values are saved in the application's option section when the window is closed as defaults for the next session.

Values are, in order:

1. 0 for off or non-zero for on.
2. Font width in pixels when debugging autosizing is off.
3. Font height in pixels when debugging autosizing is off.
4. Application display size in columns (usually 80 or 132).

Value: Four numeric values separated by blanks.

Default: 1 0 0 80

Example: DebugAutoSize= 0 7 12 80

HardMode

Purpose:

Specify whether the Hard Mode option (from the Debug Window's Options menu) is active. Refer to Section 1.6.1 for details on this option.

Value: Numeric, 0 = not active, 1 = active

Default: 0

Example: HardMode=0

DebugWindow

Purpose: A display preference option that specifies where and how large the Debug Window should appear. Five numbers are required, specifying the x and y coordinates (in pixels) from the top left of the screen, the x and y size of the screen (in pixels) and a fifth number showing whether the window is initially visible.

NOTE: If the number of pixels is out of range for the display being used (i.e., set to 790 when the monitor can only display 748 pixels horizontally), the Debug Window is automatically moved to make it at least partially visible.

The location, size and visibility of the Debug Window may be changed by the user. The values are saved in the application's RTIWIN.INI option section when the window is closed as the defaults for the next session.

Value 5 Numeric

1. Specifies x pixels from the top left of the screen.
2. Specifies y pixels from the top left of the screen.
3. Specifies x horizontal size in pixels from 1.
4. Specifies y vertical size in pixels from 2.
5. 1, Debug Window is visible, 0 it is not visible.

Default: Defaults provided by system

Example: DebugWindow = 1 3 640 271 0

1.6.3 $OSERR Enhanced

As of Revision 4.10, file sharing has been extended to better support multi-tasking environments when SHARE is installed (e.g., DOS tasks under MS-Windows, DESQview, Novell DOS 7, etc.) and mixed networks.

To better diagnose any sharing conflicts that may occur under NetBIOS networks and multi-tasking environments, the messages reported for MS-DOS errors ($OSERR) have been extended to include the extended MS-DOS error codes 13-58 (this was previously done only on the MS-Windows implementation of NPL).

NOTE: Developers with translated RTIIERR.HLP or RTIPERR.HLP files should update their files to include these new error messages.

1.6.4 LIST DT Enhanced

As of Revision 4.10, the LIST DT command has been revised to indicate special situations related to file locking. Files that are determined to be globally read-only (and so do not require file locking on a $OPEN), appear with the following designation:

```
$OPEN/xxx      :; Read only file
```

Files that are determined to be local, where there is no SHARE loaded (so file locking is not available), appear with the following designation:

```
$OPEN/xxx      :; local file, no SHARE.EXE
```

NOTE: These two cases represent the only legitimate cases where two RunTimes may have the same diskimage file $OPEN at the same time.

WARNING -- Applications that share local files must have SHARE loaded to avoid possible data corruption. If SHARE is not loaded, file locking of local files is effectively ignored. MS-Windows can issue a warning message box before permitting a second Windows RunTime to access local files when SHARE is not loaded, but does not detect DOS tasks accessing the file in similar circumstances.

Raw diskettes are an exception when considering diskimage locks. In general, diskimage locks on raw diskettes cannot be enforced under MS-DOS. The MS-Windows version of the RunTime contains logic to prevent two MS-Windows tasks from accessing the same raw diskette at the same time, but this mechanism is not respected by DOS tasks running at the same time.

1.6.5 HELP Display Enhanced

The HELP display now expands any ASCII tab characters it encounters in source text to an equivalent number of spaces in a source ASCII file (tabs at 8-character tab stops). The warning generated by the Indexed Help file processing utility:

```
Warning: TAB character(s) in source file.
```

can be ignored, if the source editor used to create the file uses 8-character tab stops.

NOTE: The resulting indexed HELP file can only be used with Revision 4.10 of NPL or greater.

1.6.6 MAT SORT Improvement

As of Revision 4.10, performance of MAT SORT under 386/DOS-Extender versions of NPL for arrays less than 64K bytes and with 2-byte locators has been improved to the equivalent of Release III.

NOTE: Arrays larger than 64K bytes are still permitted with 2-byte locators, but are slightly (estimated 15%) slower than would be predicted based on performance for smaller array sizes.

1.7 New and Updated System Variables

This section discusses all NPL Revision 4.10 enhancements to the $OPTIONS and $MACHINE system variables. Refer to the NPL Release IV Statements Guide, $OPTIONS and $MACHINE, respectively, for a complete description of these system variables.

1.7.1 $OPTIONS

The NPL RunTime allows for the inspection or modification of the $OPTIONS system variable, which consists of a variety of options. Bytes which have been enhanced or added under Revision 4.10 of NPL for MS-DOS based environments are listed below.

Byte 14 - Implied $BREAK after Disk I/O

$OPTIONS byte 14 is no longer ignored under MS-DOS. In a multi-session environment (e.g., MS-DOS tasks under MS-Windows) setting this byte to HEX(01) improves interleaved access to files. The default remains HEX(00).

Byte 42 - Forced RETURN ERROR(x)s

The following new bits in $OPTIONS byte 42 are defined for restricting the use of statements which would cause forced RETURN ERROR(x). These new bits are only applicable for operations due to attempts to clear the return stack during a callback from an external routine or functions called while program resolution is in progress.

NOTE: HEX(01) and HEX(02) bits have been previously assigned. Refer to the NPL Release IV Statements Guide, $OPTIONS, for a complete description of these bits.

Byte 42	HEX(04) bit = 0	Statements which attempt to clear the return stack frame during a callback from an external library generate a forced RETURN ERROR(251) to the external caller.
	HEX(04) bit = 1	Statements which attempt to clear the return stack frame from a callback from an external library generate an error 264. The return stack is not affected.
	HEX(08) bit= 0	Statements which attempt to clear the return stack frame during a function call used in a declaration (while a RUN statement is in progress) generate a forced RETURN ERROR(262) to the declaration.
	HEX(08) bit=1	Statements which attempt to clear the return stack frame from a callback from an external library generate an error 265. The return stack is not affected.

HINT: These values may be set by developers to ensure that unstructured exits are handled differently (when the forced RETURN ERROR action is inappropriate).

NOTE: If either of these options is set and the function call encounters an error, it is necessary to inspect the current call stack (using LIST STACK) and provide appropriate return values for functions using immediate mode RETURN(x) or RETURN ERROR(x) commands. This is true until eventual program execution returns to the external callback or declaration expression.

Statements which would clear the return stack (e.g., CLEAR P, program modifications, LOAD statements or RUN statements) are effectively disabled by these options until the special call conditions are completed.

In addition to manually setting these $OPTION bits, several function calls have been bundled with the NPL Gateway to MS-Windows API Library, which can be used to set or clear these bits whenever necessary. Refer to the NPL Gateway to MS-Windows Programmer’s Guide for details.

Byte 46 - Operation of Keys in the HELP Display

This byte is documented in the NPL MS-DOS Addendum, but not in the NPL Release IV Statements Guide. The description is repeated here as a convenience to developers.

Byte 46	Allows customization of the HELP processor's use of keys. The allowed values for Byte 46 are:	
	HEX(00)	RETURN key advances to next HELP field (default).
	HEX(01)	RETURN key operates like EXECUTE on the current HELP field.

Other values are reserved and should not be used.

Byte 47 - Windows Control

As of Revision 4.10 of NPL for MS-DOS environments, the following bits in $OPTIONS byte 47 are defined for controlling the functionality of the NPL MS-Windows RunTime.

Byte 47	HEX(01) - bit = 0	RTIWIN may poll message queue to check for HALT key and allow other tasks to run during extended operations.
	HEX(01) - bit = 1	RTIWIN never polls message queue to check for HALT key or allow other tasks to run during extended operations.

NOTE: When the HEX(01)-bit option is set, the NPL application runs exclusively between operations which wait for input (KEYIN, INPUT, LINPUT, STOP) or elapsed time (delays due to SELECT P). It normally cannot be HALTed unless a separate Debug Window is active. Menu options, including windows resizing, etc., are disabled between these operations.

Byte 47 (cont'd..)	HEX(02) - bit = 0	The RTIWIN output for Immediate mode operations can appear in the same window as the application, if the current DebugWindow option is disabled.
	HEX(02) - bit = 1	The RTIWIN output for Immediate mode operations should always appear in a separate DebugWindow. If the Debug Window option is off, the Debug Window may not appear until the first Immediate mode operation occurs.
	HEX(04) - bit = 0	If a STOP statement occurs while processing an intertask message dispatched by SendMessage, a warning indicating a ReplyMessage() / Yield() has been issued to avoid a deadlock condition is displayed. MS-Windows may hang at this point.
	HEX(04) - bit = 1	If a STOP statement occurs while processing an intertask message dispatched by SendMessage, NPL responds as if an EXECUTE key had been pressed to continue execution. Warning message #4, "STOP during InSend Message. Automatic execute option used." is displayed.

NOTE: The HEX(04), (08), and (10) bits may be set by event-driven applications which use the NPL Gateway to MS-Windows API library and have their own fetch/dispatch loop. The task switcher mechanism may use intertask SendMessage calls to activate applications. Caution must be exercised if STOP statements are embedded in NPL code that might be called during processing of the following message values.

Message ID	**Message Number**
WM_ACTIVATE	6
WM_SETFOCUS	7
WM_KILLFOCUS	8
WM_GETTEXT	13
WM_ERASEBKGND	20
WM_ACTIVATEAPP	28
WM_ICONERASEBKGND	39
WM_?? (UNDOCUMENTED)	51
WM_WINDOPOSCHANGING	70
WM_WINDOWPOSCHANGED	71
WM_NCPAINT	133
WM_NCACTIVATE	134
WM_?? (UNDOCUMENTED)	136
WM_?? (UNDOCUMENTED)	783

Byte 47 (cont’d.)	HEX(08) - bit = 0	The "Debug" option of the main window is enabled.
	HEX(08) - bit = 1	The "Debug" option of the main window is disabled.
	HEX(10) - bit = 0	NPL’s internal memory reorganization is enabled.
	HEX(10) - bit = 1	NPL’s internal memory reorganization is disabled.

All other bits are reserved. The default value is 0.

NOTE: NPL’s internal memory reorganization, which can cause addresses returned by the NPL Gateway to MS-Windows API calls ’CastLPSTR() and ’CastLPCSTR() to become invalid before they are used, may be explicitly disabled by using the HEX(10) option bit.

In addition to manually setting these $OPTION bits, several function calls have been bundled with the NPL Gateway to MS-Windows API Library, which can be used to set or clear these bits whenever necessary. Refer to the NPL Gateway to MS-Windows Programmer’s Guide for details.

NOTE: $OPTIONS is a 64-byte variable. For a complete description of all bytes within the $OPTIONS system variable, refer to the NPL Statements Guide, $OPTIONS.

1.7.2 $MACHINE

The NPL RunTime allows for the inspection or modification of the $MACHINE system variable, which consists of a variety of options. Bytes which have been enhanced or added under the Revision 4.10 of NPL for MS-DOS based environments are listed below.

Bytes 30 and 31 - Reserved

These bytes are used by NPL internally and should not be used by NPL developers.

Byte 32 - RunTime Environment Detected

Byte 32 of $MACHINE is used to record the networking operating environment detected by NPL at startup of the RunTime. Bits in byte 32 of $MACHINE have the following meanings:

Byte 32	HEX(01) - bit = 0	Security was passed from a non-local drive or from a network-wide once-per-day security mechanism.
	HEX(01) - bit = 1	Security was passed from a local drive (i.e., local drive or Gold Key diskette). When no NIAKNETB or Novell NetWare network is detected, it enables full file sharing capabilities that would otherwise not be available. In addition, this would indicate that the current user has not been charged against the network-wide user limit (only if a Novell NetWare network is detected).
	HEX(02) - bit = 0	The NIAKNETB TSR was not detected. Security passed from a local drive (i.e., local drive or Gold Key diskette). This does not charge against the network-wide user limit.
	HEX(02) - bit = 1	The NIAKNETB TSR was detected.
	HEX(04) - bit = 0	A Novell NetWare network environment was not detected.
	HEX(04) - bit = 1	A Novell NetWare network environment was detected.
	HEX(08) - bit = 0	This is a NetBIOS RunTime.
	HEX(08) - bit = 1	This is a Niakwa Networks RunTime.

All other bits are reserved for future use. The default value is 0.

Byte 33 - File Sharing and Count Methods Enabled

The high order nibble of $MACHINE byte 33 is used to record the user count method used by NPL for the current user. The low order nibble of byte 33 is used to record the locking mechanism used by NPL for the current user.

Byte 33	HEX(0F) - bit = 00	Network files are not shareable.
	HEX(0F) - bit = 01	Network files are locked by SHARE.
	HEX(0F) - bit = 02	Network files are locked by Novell NetWare calls.
	HEX(F0) - bit = 00	Uncounted RunTime.
	HEX(F0) - bit = 10	Counted from local fingerprint.
	HEX(F0) - bit = 20	Counted using NIAKNETB.
	HEX(F0) - bit = 30	Counted using a Novell NetWare semaphore.
	HEX(F0) - bit = 40	Counted using IBM AS/400 security.

All other bits are reserved for future use.

NOTE: $MACHINE is a 64-byte variable. For a complete description of all bytes within the $MACHINE system variable, refer to the NPL Statement' s Guide, $MACHINE.