



UpBoard/Pick Version 91a Release Notes

Contents

| | |
|----------------|---|
| Overview | 1 |
|----------------|---|

New Features and Enhancements

Boot Up Enhancements

| | |
|---------------------------------|---|
| Autoboot | 2 |
| Full Memory Test | 2 |
| PC Extended Memory Enable | 2 |

Peripheral Enhancements

| | |
|--|---|
| Multiple Tape Units | 3 |
| Peripheral Device Default Settings | 3 |
| Console Keyboard | 3 |
| Floppy Disk Format..... | 4 |
| Streaming Cartridge Tape Drives | 4 |
| 1/2" Tape Drive Support | 4 |

DATA/BASIC Enhancements

| | |
|--|---|
| CHAP | 5 |
| Compiler | 5 |
| CONNECT and DISCONNECT | 6 |
| CONVERT | 6 |
| Dynamic Arrays | 6 |
| FIELD() | 6 |
| FMT() | 7 |
| FOLD | 7 |
| INPUT with ELSE clause and TIMEOUT | 7 |
| MATCHES | 7 |
| MAXIMUM() and MINIMUM() | 8 |
| PRECISION | 8 |
| READTX | 8 |
| SLEEP Enhancements | 9 |
| SOUNDEX | 9 |

| | |
|---|-----------|
| STATUS() | 9 |
| SWAP | 10 |
| SYSTEM() Enhancements | 10 |
| TIMEOUT | 13 |
| TRIM() Function supports Additional Parameters..... | 14 |
| WRITETX | 14 |
| BASIC Debug Enhancements | 15 |
| EDITOR Enhancements | 15 |
| TCL Verb Enhancements | |
| ACCESS (S) Option | 19 |
| DISPLAY-BREAK | 19 |
| INPUT-PARITY | 19 |
| LIST-LINES | 19 |
| LISTPEQS (H) Option | 20 |
| PVERIFY | 20 |
| SET-BAUD..... | 20 |
| SET-FLOPPY | 21 |
| SET-LOGON Verb | 21 |
| SET-PORT-OFF Verb | 21 |
| SET-PORT-ON Verb | 21 |
| SET-SCT | 22 |
| SP-ASSIGN (M) Option | 22 |
| TA-OFF Verb | 22 |
| TA-ON Verb | 22 |
| T-SELECT Verb | 22 |
| CREATE-ACCOUNT Logon PROC | 22 |
| New GLOBAL.COMPILE Program | 22 |
| Console Function Key Support | 23 |
| DOS Services for the UpBoard | |
| DIR | 25 |
| DEL..... | 26 |
| MKDIR | 27 |
| RMDIR | 28 |
| Resolutions to Software Problems | 29 |
| Known Software Problems | 36 |

OVERVIEW

SECTION 1

Customer Service Assistance

Please read these Release Notes carefully. If you have any questions concerning this software release, our Customer Service personnel will be happy to assist you. You can call our office at (206) 821-5433 during working hours, which are 8:00 a.m. to 5:00 p.m. Pacific Standard/Daylight Time.

Conventions Used in This Document

We use the following conventions for indicating command line format:

| <u>Convention</u> | <u>Usage</u> |
|-------------------|---|
| BOLD CAPS | Anything shown in large bold characters must be typed exactly as shown. |
| <i>italics</i> | Anything shown in italics is variable information for which the user provides a specific value. |
| () | Parentheses must be typed. It is usually sufficient to type only the first parenthesis; the second is optional. |
| [] | Anything shown enclosed in square brackets is optional. The square brackets themselves are not typed. |
| < > | Anything shown enclosed in arrow brackets indicate keyboard characters. |

All punctuation marks included in the command line format (e.g., commas, parentheses, angle brackets, underscores, hyphens) are required in the format unless otherwise indicated. Square brackets are not typed.

NEW FEATURES AND ENHANCEMENTS

Section 2

The following are new features and enhancements that are in UpBoard/Pick Version 91a, and have been added to the UpBoard Application Environment.

Boot Up Enhancements

Autoboot

At boot time, if you type **PICK /A**, the message

Beginning autoboot coldstart.

is displayed. If you override the autoboot by hitting the space bar, the following prompt is displayed:

(C)oldstart/(S)system Initialize

This feature allows the system to boot directly to PICK when the computer is powered on by including the "**PICK /A**" command in the **AUTOEXEC.BAT** file.

Full Memory Test

At boot time, if you type **PICK /M**, the system will check every byte of board memory. This test take approximately four to fifteen minutes to run, depending on the amount of on-board memory installed. Pressing any key during the test will abort the test and allow the bootup to continue. The default memory test, which tests 1 byte per every 16K of memory, now takes only a few seconds.

PC Extended Memory Enable

UpBoard/Pick can now take advantage of PC extended memory (up to 8MB) for disk caching purposes on 386-based computers. At boot time, if you type **PICK /EX**, the system will enable this feature.

NOTE: Since extended memory is not used on XT's or 286-based machines, the /EX option is ignored at boot time.

The above commands switches allowing autoboot, full memory test, and extended memory enable, can be used together on one command line. The following is a valid boot command:

PICK /A /M /EX

and specifies the system to "autoboot" with a "full memory check" while enabling the use of "PC extended memory". Please note that a space must separate each option on the command line.

Peripheral Enhancements

Multiple Tape Units

The CONFIGURATOR now allows you to define one half-inch tape drive and one cartridge/cassette tape drive for use with the system.

Peripheral Device Default Settings

The default settings for all peripheral devices defined within the CONFIGURATOR now conform to those used by the PICK PC implementation.

Console Keyboard

Keystrokes generated on the system console are handled by the BIOS through the PC character buffer. This particular feature has allowed several enhancements to the system:

- * The console keyboard can be remapped through the DOS foreign keyboard mapping functions.
- * Function Keys F1 through F12 are now compatible with the PC-PICK R83 default set. The arrow keys also deliver the same codes as the current PICK PC implementation.

- * Pressing <CTL/ATL/DEL> will not force a reboot while UpBoard is running. It only echoes a BELL instead.
- * The <PRINTSCREEN> key has been allowed to pass through and will print the console screen. This is useful feature when producing documentation, but will, however, affect the other PICK users while printing.
- * DOS-based "Terminate and Stay Resident" (TSR) programs can now be "hot-keyed" into the foreground while PICK is running.

NOTE:

When the TSR is in the foreground, all PICK processes will halt until the TSR is "hot-keyed" back into the background.

Floppy Disk Format

The T-FORMAT command is now compatible with the DOS FORMAT command. T-FORMAT allows you to format floppies from UpBoard/Pick and use them under DOS. If you attempt to boot from a PICK-formatted floppy, it will give you the standard DOS unbootable diskette message rather than silently hanging.

NOTE:

When you write PICK data to a diskette, you will lose the ability to use that diskette under DOS until it is reformatted.

Streaming Cartridge Tape Drives

T-ERASE and T-RETENTION are now supported on streaming cartridge tape drives.

Pertec Tape Drive

The Pertec 9 track, half-inch tape drive is now

supported. In addition to the Cipher F880 and M990.

DATA/BASIC

BASIC Runtime Speed Enhanced

The performance of BASIC compilation and execution has been improved substantially. These changes require that all BASIC code or from foreign machines be re-compiled.

The following commands have been added changed or enhanced.

CHAP

This command allows you to set absolute priority, usually at zero. Setting it at 1 means that your job is activated only one time in two. Setting it at 2 means your job is only activated one time in three. To change CHAP within BASIC use the command:

CHAP *n*

where *n* is a number between 0 and 32. For example:

CHAP 0

removes CHAP completely for your process, activating it every time.

COMPILER

We have added a new option (F) to the BASIC compiler. This option will prompt you for the name of the file where the compiled BASIC program is to be stored. It is generally a good idea to make this file a dictionary-only file, so that you will not have to enter 'DICT filename' when you are prompted for the name of the destination file. This feature should make it much easier to generate object-code-only applications.

CONNECT and DISCONNECT

A feature of UpBoard PICK BASIC that allows use of a serial port which is not currently active (i.e., logged off). It allows the process to both output and receive input from the connected port.

CONVERT

Format:

NEW = CONVERT *original, string1, string2*

This new intrinsic function will return the *original* string with each character found in *string1* converted to the character in the same position in *string2*. It is perfectly acceptable for *string2* to be null, or a difference length than *string1*. Where there is no corresponding character in *string2*, the character in *string1* is deleted. Additional characters in *string1* are ignored.

DYNAMIC ARRAYS

Dynamic array indices larger than 32K are now supported.

FIELD()

A optional fourth parameter has been added to the FIELD statement. This parameter specifies the number of groups to be returned. Specifying one as the fourth parameter is equivalent to omitting it. Thus if we have the variable string contain:

string = ABC*DEFG*HUKL*MNOPQ*RSTU

NEW = FIELD(string, '*', 3, 2)

the string NEW would contain HUKL*MNOPQ

FMT()

In order to make this system more compatible with other PICK vendors, we have allowed format masks to also be invoked using the following format:

REPLY = FMT (variable,mask)

Note that this generates exactly the same code as:

REPLY = variable, mask

This feature is provided to make conversions easier. We cannot guarantee that masks from a different system will work the same way on this system, though this is usually the case.

FOLD

Format: **FOLD (string,width)**

FOLD performs a function similar to the T code used in ACCESS dictionaries (AMC 9). The string is broken into multiple lines (if necessary) so that each line is no longer a specified length. Each line is split at a blank (if possible). The string generated consists of lines separated by attribute marks at the fold point.

INPUT

We have enhanced all BASIC INPUT statements to support an ELSE clause. This ELSE clause will be taken if the program times out waiting for input.

To set the number of seconds before a timeout occurs, use TIMEOUT either as a BASIC command, or as a TCL verb.

MATCHES

The MATCHES operator has been enhanced to allow multiple matches to be tested at the same time. To achieve this the matching pattern must

consist of a multi-valued list of patterns. If the string being tested matches any one of them, a match has occurred. For example, to test whether a number is 4 or 5 digits long, you can now enter:

```
IF NUMBER MATCHES 4N":CHAR(253):"5N"THEN...
```

It is also now possible to test for a particular pattern not matching. For example, to test for a number not being 6 digits long you can now say:

```
IF NUMBER MATCHES "" 6N"THEN....
```

MAXIMUM() and MINIMUM()

Format: **MAXIMUM**(dynamic array)
MINIMUM(dynamic array)

We have added **MAXIMUM** and **MINIMUM** functions to return those value when presented with a dynamic array containing multiple fields. Non-numeric elements in the dynamic array are treated as zero.

PRECISION

We have enhanced the BASIC Run Time System to additionally support **PRECISION 5** and **PRECISION 6** arithmetic. This will allow you to keep up to six significant digits of precision after the decimal point in your BASIC computations. Precision 4 remains the default.

READTX

Sometimes you need to be able to process a tape written by a non-PICK system which contains segment marks (x'FF') embedded in the data records. You cannot do this with a normal **READT** command. It reads in the entire record, but then truncates the data at the first segment mark encountered.

We have added a **READTX** command to solve this problem. It converts each byte of data to a two-byte

hexadecimal representation as it stores it in the variable. This means that segment marks can now be processed as they are represented by consecutive F characters. To convert between a normal and hexadecimal character string, you can use the "M/X" conversion code, which is a standard part of the application environment.

SLEEP ENHANCEMENTS

The functionality of the SLEEP statement in BASIC has been enhanced. Sleeping for smaller time periods (hundredths of a second) is now permitted. Thus SLEEP 0.25 will sleep for approximately one quarter of a second, (usually slightly longer).

SLEEP 0 is now a special case. It has reverted to the original meaning of RQM, and gives up the current time slice. Thus it takes only a few milliseconds on a system with a few active users, and may take several seconds on a system with many active users. Sleeping until a specified time is also supported. Thus SLEEP '05.00' will sleep until 5 a.m.

NOTE: You must include the single quote marks or the wrong time will be interpreted. For example, SLEEP 9:51 would sleep for 951 seconds; SLEEP '9:51' would sleep until 9:51 a.m.

SOUNDEX

Format: SOUNDEX (string)

This new BASIC function returns the equivalent soundex value for the specified string. It uses the same algorithm as the LIKE operator which has been added to ACCESS for soundex matching.

STATUS()

In order to improve Prime Information compatibility with our implementation, we have added a new intrinsic function to BASIC called

STATUS() which does exactly the same as a SYSTEM(0). This will allow many Prime programs to run without modification.

SWAP

NEW = SWAP(Original,String1,String2)

This new intrinsic function will return the original string with every occurrence of String1 swapped to String2. It is perfectly acceptable for String2 to be null, or a difference length than String1.

SYSTEM(number) Enhanced

Format: SYSTEM(number)

The SYSTEM function in BASIC can be used to get any of the following data:

Number/Action

- 0 Return information about a previous statement.
See below.
- 1 1-Output is being sent to printer
0-Output is being sent to CRT
- 2 Current page width in columns
- 3 Current page depth in columns
- 4 Lines remaining on current page
- 5 Current page number
- 6 Number of lines printed on current page
- 7 Current terminal type(A=ANSI,V=View-point etc.)
- 8 Current tape block size
- 9 Current CPU usage in milliseconds

NUMBER/ACTION

- 10 1 - if input is being taken from stack
0 - if input is being taken from terminal
- 11 1 - If SELECT active
0 - If SELECT is NOT active
- 12 Time in milliseconds
- 13 RQM - True Release Quantum - Does NOT Sleep - Returns 1
- 14 Number of characters waiting in input buffer
- 15 Current Options (A-Z)
- 16 EXECUTE level (0 means program is not being EXECUTED)
- 17 Error message numbers(s) from last EXECUTE statement. Also, immediately after printing first line of a document to the spooler entry number as:

099: Attribute mark:Entry#
- 18 Port number for this process
- 19 Account name
- 20 User name (Same as Account name at present)
- 21 System Privilege level (0,1,2 or 3)
- 22 Current SP-ASSIGN options, forms queue and number of copies (separated by attribute marks)
- 23 System serial number (currently zero)
- 24 Maximum frame number on system
- 25 Total frames in overflow table

NUMBER/ACTION

- 26 Linked frames in overflow table
- 27 Ports configured
- 28 1 - If background process, 0; otherwise 29
0 - If no file save currently active, else port
number of save process + 1
- 30 Current CHAP value for this port.
- 31 Host system type (DOS)
- 32 Current host processor ("68K", or "88K")
- 34 Number of characters in local input buffer

SYSTEM(0) returns the following information:

- * After a READT, WRITET etc., 1 indicates tape unit is not attached if program takes the ELSE clause.
- * After a READT, 2 indicates that an EOF was read when the tape takes the ELSE clause.
- * After WRITET, 3 = attempted to write of a null string; 11 = attempted write of a record longer than current tape block size.
- * After a READT, WRITET etc., 4 indicates the tape was not loaded, 5 that the tape was write protected, and 6 that a parity error has been detected.
- * After a READT is successful and does not take the ELSE clause, the value of SYSTEM(0) is zero if the record just read from tape is the same size as the current tape block size, or the size of the record just read from tape in bytes.
- * After a LOCKED clause in a READU has been executed, SYSTEM(0) returns the number of the port which has the item locked.

- * If the ELSE clause of a LOCK statement is taken, SYSTEM (0) returns the number of the port which has the lock set.
- * After a ICONV(date, 'D') type of conversion, SYSTEM (0) is zero if the conversion worked, and 1 if the date being converted was invalid.
- * After opening a file, SYSTEM(0) is 2 if the file was access protected; - 1 if the file is update protected, and will not be able to be updated; zero if the file will also be able to be updated.
- * After attempting a READ type statement (except when the LOCKED clause of a READU/ MATREADU is active), SYSTEM(0) is - 1 if the file is update protected; zero if the file will be able to be updated.
- * If the item you are attempting to read is a binary item which cannot be processed using BASIC, the ELSE clause is taken, and SYSTEM(0) is set to 10 to distinguish this case.
- * After failing a CONNECT and taking the ELSE clause, SYSTEM(0) indicates the reason for the failure as follows:
 - 2 indicates the port was logged on.
 - 3 indicates we were already connected to a different line.
 - 4 indicates an invalid line number was specified
- * A value of zero or higher is the line number of the port currently connected to the line, preventing us from connecting.

TIMEOUT

We have added a new TIMEOUT command to BASIC. This command specifies the number of seconds the system will allow when waiting for terminal input before a timeout condition occurs.

The syntax is:

TIMEOUT n

where n is the number seconds. To turn off this feature you need to specify zero seconds. When a timeout occurs, (or if DTR drops on a line where DTR detection is enabled), the port where it occurs will normally log off. This can be prevented in BASIC by using an ELSE clause on the end of INPUT statements. If such a clause exists, it will be taken in preference to logging the terminal off.

TRIM() Function supports Additional Parameters

The BASIC TRIM() function can now support one, two, or three parameters, using the following syntax:

TRIM(string{trimchar{type}})

The second parameter, if present, can be used to specify a character other than spaces to be trimmed by this function. The third parameter, if present, can take any of the following values:

- L - trim leading
- T - trim trailing
- B - trim leading and trailing
- A - trim all
- R - trim redundant

TRIMB(string) and TRIMF(string)

These two new BASIC functions are very similar to the existing TRIM() function.

- TRIMB() only removes trailing spaces.
- TRIMF() only removes leading spaces

WRITETX

Sometimes you need to be able to produce a tape for a non-PICK system which contains segment marks (x'FF') embedded in the data records. You cannot

do this with a normal WRITET command. It truncates the record at the first segment mark it encounters.

We have added a WRITETX command to solve this problem. It takes each byte of data from a two byte hexadecimal representation as it is stored in the variable. This means that segment marks can now be processed as they are represented by two consecutive F characters.

To convert between a normal and hexadecimal character string, use the "M/X" conversion code which is a standard part of the operating system.

BASIC DEBUG ENHANCEMENTS

C - displays the line of source code being executed when the break occurs. This is extremely useful when single stepping your program. It will not work if you have moved your source code or object code, unless you use a 'Z' command to specify where the source code can be found.

M - allows a toggle of the modal trace feature. If it is on, the BASIC debugger will be entered each time the programs executes a CALL (C), RETURN (R), or EXECUTE (E) statement.

Q - displays the current program name when a breakpoint is encountered.

S - displays the subroutine return stack showing the calls used to get to this point in the program.

? - added time and date of compilation.

EDITOR

Appending Text To The End Of A Line

The **A** command will now allow you to append text to the end of an existing line. **A/ABC** will append the string **ABC** to the end of the current line.

A4/ABC will append the string **ABC** to the end of the current line and the next three lines.

Exit Verification

The editor will now ask 'Are you sure?' for a confirmation before exiting a changed item.

Breaking Lines Into Two

The **B** command will now allow you to break a line in two. **B20** will break the current line at column 20. **B/ABC** will break the current line at string **ABC** with the **ABC** remaining on the end of the first line.

(L) No Longer Required by Locate Command

It is now possible to omit the **L** at the front of a locate command when using the editor. **L/ABC** and **/ABC** will give the same result, as will **L/ABC** and **ABC**.

Defining A Block To The Editor

It is possible to define a block to the editor, and then use that block in subsequent move, copy and delete commands. To set the start of the block use the *less than* (<) command. To set the end of the block use the *greater than* (>) command. Use the question mark (?) command to see the current block setting. The end of block marker cannot be at a line less than the start of block marker. If the start of block is moved below the end of block marker any movement command will give the 'Block not defined' message. Any block definition is dropped when the **F** command is used, or by using the new **BU** command.

Copying A Block In The Editor

If you have a block currently defined, entering the **BC** command will make a copy of that block at the location where the cursor is currently positioned.

Possible error messages

Block not defined
SEQ

Deleting A Block In The Editor

If you have a block currently defined, entering the **BD** command will delete that block. You will be asked to confirm the deletion with the following display:

Type Y to delete block from _ < startline _> to _
< endline _>

Starline is defined by the start block marker and
endline is defined by the end of block marker.

Possible error messages:

Block not defined
SEQ

Displaying And Replacing Line In Hexadecimal

We have added a new **H** command which works exactly like the **L** command except that it displays lines in hexadecimal notation. This if **L2** displays:

```
002 HTIME  
003 P
```

then **H2** will display:

```
002 4854494D45  
003 50
```

In order to replace characters using hexadecimal character strings use the **RH** command which works exactly like the **R** command except that the string to be replaced, and the replacement string, must both be valid hexadecimal character strings.

The **RH** command supports the various additional syntaxes of the **R** command. Thus **RHU9999/09/2020202020/** is a valid command to replace all occurrences of a tab character by six spaces. Note that when using this command, as with the

standard R command, a trailing hex '20'(blank) be followed by the closing delimiter.

Joining Lines Together

A new **J** command has been added to the editor which will join two (or more) lines together. It works by joining the number of lines specified (default 1) below the current line onto the end of the current line. For example, **J3** will join lines 11 thru 13 onto the end of line 10 if the cursor is positioned on line 10 when the command is issued.

Moving A Block In The Editor

If you have a block currently defined, entering the **BM** command will delete that block from its original location and move the block to the location where the cursor is currently positioned. This command is equivalent to a block copy of a subsequent block delete of the original data.

Possible error message:

```
Block not defined  
SEQ
```

Using the TAB Key In The Editor

The **TAB** key now works differently. The only place in the operating systems where it generates spaces is in the editor. There the default is set to six spaces for the first tab, then three spaces for each subsequent tab. The **C** command, which displays a columnar ruler, has been modified to display the current tab settings as part of the ruler.

Undo Block Defining Parameters

To remove the currently defined block pointers, (i.e., the startblock and endblock pointers), enter **BU**. This does not change or move any data in the item; it only removes the pointers or definition of the current block. Attempting to use the block move, block copy, or block delete command after

using this command will produce the "Block not defined" message.

Default P Setting

We have changed the setting for P when you initially enter the editor. Formerly it was always set to list 22 lines. Now it will list the number of lines specified by the TERMN statement (parameter 2) less 2. This is still 22 if your TERM statement is, for example:

```
TERM 79,24,0
```

It now gives a much better default when using a 60 line terminal. It also allows you to determine the default using the TERM statement.

TCL

ACCESS (S) OPTION

An (S) option has been added to the ACCESS verbs to suppress the "item not on file" messages at the end of a listing.

DISPLAY-BREAK

New name for previous DISPLAY-KEY verb.

INPUT-PARITY

We have added a range parameter to the INPUT-PARITY verb. The syntax for the range parameter is:

```
INPUT-PARITY ON (start-end) or
```

```
INPUT-PARITY OFF (start-end)
```

LIST-LINES

LIST-LINES lists the hardware setups for serial I/O ports. The syntax is:

```
LIST-LINES (start-end)
```

LIST-LINES can be used to display the current baud rate, parity, flow control, and other information of any or all ports on your system. If no parameters are specified, all lines are display. If dashes are displayed in the baud rate column, then no physical device is associated with the corresponding port. If the physical device for a port has no "baud rate", a word (such as "Pptr" or "Smile" or "Net") will appear in the baud rate column to denote the type of device. If only one parameter is specified, only that line is displayed.

A device column to show the device path for connected ports was added.

LISTPEQS H Option

The **LISTPEQS H** option displays only on the hold entries in the printer que. This option is not documented in **UpBoard User Manual**.

PVERIFY

Time & Date stamp of last compile has been added to the **PVERIFY** output.

SET-BAUD

The syntax for the **SET-BAUD** verb has been extended to allow:

- * changing the settings for a range of ports
- * specifying the parameters in any order
- * using ", " or SPACE to separate parameters

Examples of acceptable syntax are:

SET-BAUD 6,9600 (E
SET-BAUD 6-12,9600,EVEN
SET-BAUD 19200 ODD (21-25

SET-FLOPPY

The **SET-FLOPPY** command has been added for compatibility. Options with this command are A & B for drive selection and S & H for standard and high density.

SET-LOGON VERB

A SET-LOGON verb has been added to the SYSPROG account to allow you to change the number of logon attempts allowed before keyboard input is inhibited.

The syntax of the command is:

SET - LOGON *n*

where "n" equals the number of logon attempts allowed and must be a number between 0 and 255. An entry of "0" (zero) will allow an unlimited number of logon attempts. SET-LOGON with no number will display the current status.

SET-PORT-OFF VERB

A new SET-PORT-OFF verb has been added that will turn off UpBoard/PICK dedicated ports. Its normal use will be with the CONNECT and DISCONNECT statements in BASIC.

The syntax for this verb is:

SET-PORT-OFF (*start-end*)

SET-PORT-ON VERB

A new SET-PORT-ON verb has been added that will turn on UpBoard/PICK dedicated ports. The syntax for this verb is:

SET-PORT-ON (*start-end*)

SET-SCT

The SET-SCT command has been added for compatibility.

SP-ASSIGN M OPTION

The SP-ASSIGN M option suppresses the "Entry n" message. This option is not documented in the UpBoard User Manual.

TA-OFF

A new TA-OFF verb has been added that will turn off the 1024 character type-ahead buffer. Type-ahead is "on" at boot time. The syntax for this verb is:

TA-OFF *(start-end)*

TA-ON

A new TA-ON verb has been added that will turn on the 1024 character type-ahead buffer. Type-ahead buffer. Type-ahead is "on" at boot time. The syntax for this verb is:

TA-ON *(start-end)*

T-SELECT

New verb to change currently assigned tape device (replaces SET-TAPE, which remains for backwards compatibility).

CREATE-ACCOUNT Logon Proc

A small LOGON proc will now be placed in newly created accounts. It will set up the symbol table so that aborts can be traced, display the UpBoard/PICK version number and display the current time and date.

NEW BASLIB PROGRAM

A new program, GLOBAL.COMPILE, has been added to the BASLIB file to assist in upgrades that require a recompilation of all BASIC code.

The program recompiles all BASIC programs with existing object code in one or more accounts. The accounts may be specified on the command line or in a SELECT list. If no parameters are present on the command line and no SELECT list is present, all accounts on the system are processed.

OPTIONS:

- (P) Send output to printer
- (V) Verify BASIC object code only
- (Z) Verify after compiling each program
- (Q) Make Q-pointers to all "D" and "Q" files in target account (use this option with great care, for example, by running GLOBAL.COMPILE from within a temporary account with SY33 privileges.)

Console Function Key Support for UPBOARD

The following programs and commands have been added to this release of UpBoard/PICK to provide console function key support for all UPBOARD computer system users.

SET.FUNC.KEY Program

The SET.FUNC.KEY program allows you to selectively program any console function key, arrow key, INSERT,HOME,PAGE UP,DELETE, END, and PAGE DOWN keys on the keyboard. Any key may be set to any value up to 255 characters.

The syntax of the command is:

SET.FUNC.KEY *function.key* *function.key.string*

The *function.key* argument must be one of the predefined mnemonics for the requested function key. The mnemonics are:

- | | |
|-----------------|--|
| F _n | Function key "n" where "n" is from 1 to 12 |
| SF _n | Shift Function key "n" |
| AF _n | Alt Function key "n" |
| CF _n | Control Function key "n" |
| UP | Up arrow key |

| | |
|----------|-----------------|
| DOWN | Down arrow key |
| LEFT | Left arrow key |
| RIGHT | Right arrow key |
| INSERT | Insert key |
| DELETE | Delete key |
| HOME | Home key |
| END | End key |
| PAGEUP | Page Up key |
| PAGEDOWN | Page Down Key |

SET.FUNC.KEY is a BASIC program in BASLIB which uses a user exit call in conjunction with a numeric key index. You may want to write your own program for setting multiple keys or change the mnemonics for compatibility with a language other than English.

SET-FUNC Verb

The SET-FUNC command is a BASIC program which emulates the PICK PC form for setting function keys. It is supported with the file FUNCKEYS and the tables DEFAULT, NULL, and TEST. It is supplied for compatibility for those upgrading from PICK PC systems to UPBOARD or ICON 2600 systems.

They syntax of the command is:

SET-FUNC *function.key.table*

FUNCTION.KEY.DEFS in BASLIB

The following list contains the function key definitions in BASLIB.

```

001 EQU AM TO CHAR(254)
002 DIM INDEX.TABLE(14)
003 EQU VM TO CHAR(253)
004 EQU SET.FUNC.DEY TO 'U8212'
005 FUNCTABLE="F".ANL"SF":AM:"CF":AM:
    "AF":AM
006 FUNCTABLE=FUNCTABLE:"UP":AM:
    "RIGHT":AM:"DOWN":AM:"LEFT":AM
007 FUNCTABLE=FUNCTABLE:"INSERT":AM:

```

```

"DELETE":AM:"HOME":AM:"END":AM
008 FUNCTABLE=FUNCTABLE:"PAGEUP":AM:
"PAGEDOWN"
009 INDEX.TABLE(1)=L;* F
010 INDEX.TABLE(2)=11;* SF
011 INDEX.TABLE(3)=23;* CF
012 INDEX.TABLE(4)=35;* AF
013 INDEX.TABLE(5)=48;* UP
014 INDEX.TABLE(6)=49;* RIGHT
015 INDEX.TABLE(7)=50;* DOWN
016 INDEX.TABLE(8)=51;* LEFT
017 INDEX.TABLE(9)=52;* INSERT
018 INDEX.TABLE(10)=53;* DELETE
019 INDEX.TABLE(11)=54;* HOME
020 INDEX.TABLE(12)=55;* END
021 INDEX.TABLE(13)=56;* PAGE UP
022 INDEX.TABLE(14)=57;* PAGE DOWN

```

DOS Services for UPBOARD

The following DOS commands are available from the UPBOARD/PICK for environment.

DIR

DIR lists files and subdirectories in a directory. The listing may include the size of each file and the date and time the file or subdirectory was created. The syntax for the command is:

```
DIR [d:] [pathname] [filename] [/P] [/W]
```

Parameters:

| | |
|----------|---|
| d: | disk drive designator indicating the disk for which you want a listing (default is the disk drive that booted UPBOARD/PICK) |
| pathname | directory path name, entered when you want to list the files and subdirectories in a specific directory (default is the directory that booted UPBOARD/PICK) |
| filename | file name, entered when you want to list all files with a specific name |

/P (print mode) outputs the directory to the printer

/W (wide display mode) displays file names in columns, five file names per line, without any other file data

Examples:

DIR A:\BOOK1 lists all the directory entries (files and directories) in directory "\BOOK1" on drive A, with one line of information per file or subdirectory.

DIR*.068 /W list file names (but no other file information) of all files in PICK directory with an "068" extension. The display is shown in wide format.

DEL

DEL deletes one or more specified files

The syntax for the command is:

DEL [d:] [pathname] filename

Parameters:

d: drive designator for disk containing the file(s) you want to delete

pathname complete path name for the file(s) you want to delete

filename name of the file you want to delete, or name with wild cards indicating multiple file to delete

Examples:

DEL C:\JUNK.TXT deletes the file JUNK.T XT from the root directory on the disk on drive C

DEL *.BAT deletes all files with the file name extension ".BAT" in the UPBOARD/PICK directory

DEL \MAIL* deletes all files in the "\MAIL" directory on the drive that booted UPBOARD /PICK (you are prompted to confirm this deletion)

MKDIR

MKDIR creates a new directory

The syntax for the command is:

MKDIR [C:] *pathname*

Parameters:

C: drive designator for the drive containing the disk on which the directory will be created (default is the disk drive that booted UPBOARD/PICK)

pathname path name for the new directory, in the path name you specify, all directory levels except the lowest must already exist; if the path name does not begin with a backslash (which indicates the root directory), the new directory will be created under the directory that booted UPBOARD/PICK

Examples:

MDIR /JOHN

creates the directory "JOHN" in the root directory of the drive that booted UPBOARD/PICK

MKDIR C:\WP\RS

creates the subdirectory "RS" under the "\WP" directory on drive C if the "\WP" directory already exists

RMDIR

RMDIR deletes a selected directory
The syntax for the command is:

RMDIR [C:] *pathname*

Parameters:

d: drive designator for the disk containing the directory to delete (default is the disk drive that booted UPBOARD/PICK.

pathname the path name of the directory to delete

NOTE

Before you can remove a directory, it must be completely empty--no files or subdirectories may still exist within the directory to be deleted. Also, the directory to be deleted may not be the directory that booted UPBOARD/PICK.

Example:

RMDIR \JOHN deletes the directory "JOHN" in the current drive's root directory.

Software Resolutions Since Release 6.12

Resolutions to Software Problems

This section provides information about resolutions that have been made to known problems in the UPBOARD/PICK for software. Since the problems span more than one release of the software, the resolutions are discussed in the appropriate "Release Version" section.

ABS Changes

- * Auxiliary printing now works using AUX-ON or SP-ASSIGN A.
- * ABS tape and MLOAD checksums now match.
- * MDUMP ASCII display now matches the HEX display.
- * Partial LOCATE will null attributes in source and "[" in target now provides the correct position for target.
- * RUNOFF does not abort with tabs set and a tab marker (" > ") at end of line.
- * BASIC compiler 32K limit for forward or backward branches no longer causes a fail during compile.
- * SYSTEM DEBUGGER -lower to upper case conversion is now performed on user input at data entry prompt.
- * READTX and WRITETX now function.
- * The BASIC NAMED COMMON, in use with random reads, no longer causes GFEs or corrupted ABS.
- * LINK-WS no longer clears spooler hold files.
- * Deleted spooler hold files are now released back to the overflow; a system restore is no longer

required to recover the space.

- * COMPUSHEET+ now handles large number divide and multiply problems.
- * BASIC CONNECT now continues to leave the connected port disabled.
- * Logon attempts after a failed password have been increased.
- * Incremental restores no longer aborts after selecting the tape device. T-LOAD can now restore file by file and full restore.
- * Floppy formats now detect when the format has failed.
- * SET-TERM will now change the executing port if it is included in the range.
- * A "control-S" character is now returned after MODEM-OFF instead of a "garbage" character.
- * BASIC CONVERT and SWAP functions now handle strings longer than 32 Kbytes.
- * The BASIC compiler no longer aborts with very large object programs.
- * A BASIC workspace overflow problem (NUM function 20000 times in MEW.BENCH) has been corrected.
- * QSELECTS of multivalued and subvalue combinations now correctly returns all elements.
- * The SAVE verb option G no longer disables the Y option.
- * The words AT and MISSING no longer appear to the compiler to be reserved words.
- * A latency problem with X-OFF character has been corrected.

- * Virtual tape no longer gives a premature end of media.
- * Executing SET-BAUD on a non-serial device no longer hangs the device.
- * LIST-LINES no longer displays invalid parameters for non-serial lines.
- * A problem in reading "virtual" tape from floppy drives has been corrected.
- * A problem with repeating characters on serial printers that filled the input buffer with "X-OFF" characters has been corrected.
- * Backwards-compatible SAVE conversions of large items will not convert items larger than 32267 characters. The online documentation for SAVE has been changed to reflect this restriction. (Refer to the following documentation in Volume II of the UPBOARD/PICK User Manual:
 - Section 10.32, "System Backup and the SAVE Verb"
 - Option C in Section 1032.2, "Options" for further detail.)

Monitor Changes

- * Multiply and divide with positive returns no longer output with a negative sign.
- * Timeout past midnight no longer causes a port to hang.

PICK.EXE Changes

- * Intermittent hangs & parity errors no longer occur on cartridge tapes.
- * Input (especially when bursting) no longer loses characters.
- * Serial I/O cards no longer report a false status,

which caused a general system hang.

- * Co-processor synchronization no longer causes intermittent system hangs.
- * The bug in reading 'virtual' tape from floppy has been eliminated.

Configurator Changes

- * PC BIOS ROM'S no longer cause erroneous interpretation of floppy drives.

Software Resolutions Since Release 6.11

The following problems were identified as existing UPBOARD/PICK 6.11 and have been resolved in this release of the software.

- * The default block size for cartridge tape devices was inadvertently left at 32768. This caused a WRITET in BASIC to fail. The default block size has been changed back to 16384.
- * While tape operations were running, key strokes from other terminals could cause a system hang. This no longer occurs.
- * Overflow multiplication problems have been corrected with this release.

Software Resolutions Since Release 6.00

The following problems were identified as existing UPBOARD/PICK Release 6.00 and have been resolved in this release of the software.

- * The Configurator now supports CGA and MONO adapter cards. We have added a text mode to allow support of certain graphics and non-graphics adapter cards. The text mode will be used automatically whenever certain cards are sensed. Text mode can be forced when running the Configurator by typing:
CONFIGURE /T
- * Daylight Savings Time is now compatible.
- * When PICK was shut down, DOS was

prevented from accessing the floppy disk.

- * Timekeeping for the realtime clock was causing a 1% slowdown. This has been corrected.
- * A problem which did not allow the attachment of floppy drives on some older BIOS' has been corrected.
- * The BIOS critical error vector is now properly trapped and the error code is passed on to PICK rather than causing the PC to crash.
- * The console keyboard no longer intermittently hangs.
- * Data spanning multiple tapes on streaming cartridge tape drives is now performed correctly
- * Using TIMEOUT in BASIC with an INPUT VAR, length caused subsequent inputs with lengths to be fixed to the first specified length. This no longer occurs.
- * EXECUTE with capturing would not output to a printer. This now outputs properly.
- * The STARTPTR verb failed to allow from queue changes once it had been performed.
- * LOGTO *accountname,password* would not accept the correct password.
- * Using prestore command P1 FD(P1 in the EDITOR did not skip the "are you sure?" question.
- * Attaching the tape unit to a block size less than 40 did not work.
- * Pressing "Control-X" cancelled an input line with a line feed.
Resolution: "Control-X" now erases all characters on the input line and suppresses

the line feed.

- * ECHO OFF in BASIC did not suppress the backspace.
- * RECALCULATE-FILE-SIZES, as run from FILE-SAVE, had the downsize file option set.
- * PVERIFY has been fixed for BASIC object code greater than 32K.
- * Error codes 1171 and 1172 for LISTPTR were backwards. They now show the correct error code.
- * Using the MSG or MESSAGE verbs with a message greater than 70 characters caused Frame 52 to not verify.
- * The (U) option to SP-EDIT and SP-KILL required a SY33 privilege level. It now only requires a SYS2 level.
- * LISTPEQS (F did not page correctly.
- * In some cases, depending on the end-of-frame layout of BASIC object code by the compiler, BASIC programs would abort at runtime, most likely with an "illegal opcode" message.
- * CHECKSUM did not display the file name and item name correctly and did not consistently return correct information at the dictionary level only.
- * If you ran out of disk space and used the "Q" option to quit out of the process that was trying to write out a record, you had to clear out all unwanted data and do a VERIFY-FILES check for GFES.
- * QSELECT of a non-existent item returned an active select list.
- * SP-KILL n-m only deleted the first printer. The same is true for the A and B options.

- * When you used the BASIC "CRT" syntax rather than "PRINT" after using the CONNECT verb, the output did not go to the local port as it is supposed to and the remote logon was not suppressed.
- * CompuSheet+ occasionally aborted while printing or copying a spreadsheet.
- * The on-line documentation for the BASIC SYSTEM(5) and SYSTEM(6) functions were reversed.
- * The default TERM parameters for the printer were inadvertently left at 80,60 instead of 132,60.
- * NAMED-COMMON variables were being initialized with random data.
- * CHOKE and IMMEDIATE option to the spooler were not functional.
- * On-line documentation for MODEM-ON, CONNECT, SYSTEM, and floating point was incorrect.
- * TANDEM lost characters under heavy I/O.

Resolution: An "S" option has been added to TANDEM to slow the port that is being "TANDEMed" to the host port. Also, use of the CHAP verb on the TANDEMed port will help.

- * Erroneous results occurred when using PRECISION 6.
- * MODEM-ON and MODEM-OFF would not allow a range of ports.

Resolution: A range option, (m-n, has been added to the verbs where an m equals the beginning port number and n equals the ending port number.

- * The SHUTDOWN message was being output to a printer ports. This no longer occurs.
- * overflow problem with ACCESS negative number multiplication has been corrected.

KNOWN SOFTWARE PROBLEMS

The following software problems are acknowledged in this release of the UPBOARD/PICK. If a work-around exists for the problem, that temporary solution has been included with the description of the problem.

- * The BASIC DELETE function has a problem following a "partial locate", i.e., the LOCATE string parameter enclosed within square brackets.
- * Intermittent parallel printer problems continue to occur when using Genacom and Printronics printers.
- * The BASIC INPUTTRAP does not function correctly in most cases. For example, when trying to trap the escape character, a print statement after the branch from the trap displays in the wrong screen position. Extra characters also appear on the screen during input.
- * BASIC variables that begin with a number (which are not valid variable names) are currently not being flagged by the compiler. Use of these variables at run time can cause unacceptable results.
- * BASIC B-Correlatives do not work with the BASIC debugger.
- * JET aborts when using Access inserts and B-Correlatives.
- * In programs with multiple SELECT TO and READNEXT loops, the SELECT TO,

followed by a READNEXT, sometimes fails and takes the ELSE clause.

- * BASIC multi-cats of more than 127 strings will cause an illegal frame abort at frame 190 during compile.
- * An old form of processing ACCESS correlatives where elements are separated by value marks may give the following abort with sufficient data:

```
CROSSING FRAME LIMIT ; REG-13 ABORT@  
90.5EE
```

A dictionary that has the following in attribute 8:

```
008 G*1]TIC.PART.MASTER.SUPP;X;;1
```

could abort when using that dictionary in an ACCESS with clause. The preferred method of writing this dictionary so the abort will not occur is:

```
008 F0; (G81); (TIC.PART.MASTER.SUPP;  
X;;1
```