



Purveyor STACKER Utility Quick Reference Guide

The STACKER utility resides on Purveyor to

- ❖ Maintain an audit trail to document destructive or mistakenly executed commands, enabling more effective problem resolution.
- ❖ Provide a more user-friendly interface than the standard utilities on jBase, including the following features:
 - Arrow key-controlled editing.
 - Storage of command stacks from session-to-session.
 - Retrieval of execution history from the audit file.
 - Multi-lined command functionality.
 - Availability of a mini-word processor for stack lines and a text editor for the entire command stack.
 - Familiar dot (.) commands.

This Quick Reference Guide serves as a handy reference tool for using the STACKER utility.

- A stack is a data area or buffer used for storing commands or requests that need to be processed.
- Each line in the stack is a stack entry (or stack line).
- A stack is always a “push-down list” (i.e., new requests are inserted at the top of the stack, “pushing” previous stack entries down the screen so that the program always processes entries from the top of the stack).

Accessing the STACKER Utility

You may access the STACKER utility in Purveyor via the following procedures:

- Type ! at any prompt and select the option for TCL.
- Select Report Writer off the menu.

*If you need TCL access but do not have it, contact your System Administrator.

Using Arrow and Shortcut Key Editing

Press the shortcut and arrow keys to display the next or preceding command in the stack entry list (for editing a line or executing a series of commands), to move within a line, or to insert and delete characters.

* If the stack entry is longer than the screen width, consider using the .C or .W commands to edit the line (see Editing Stack Entries on page 4).

- Up arrow — Displays the next command in the stack entry list.
- Down arrow — Displays the previous command in the stack entry list.
- Right arrow — Moves the cursor to the right until it reaches the end of the stack entry.
- Left arrow — Moves the cursor to the left until it reaches the beginning of the stack entry.
- Home — Positions cursor at the beginning (home) of the stack entry.
- End — Positions cursor at the end of the stack entry.
- Insert — Toggles between Replace and Insert modes.
- Del — Deletes the character “under” the cursor.

Using Control Key Functions

Control keys may be used for a variety of functions — to move the cursor position within a stack entry or to another line, toggle between two modes, access line editing help or abort the current session.

- ❖ [Ctrl-A] — Moves cursor to the beginning of the current stack entry.
- ❖ [Ctrl-B] — Toggles between Insert and Replace modes. (Default is Insert.)
 - * In Insert mode, you may enter text without overwriting the stack entry.
 - * In Replace mode, stack entries are overwritten.
- ❖ [Ctrl-C] — Aborts your current session and preserves the last command that you processed in the stack.
- ❖ [Ctrl-E] — Accesses line-editing help screens.
 - * You may exit the Help screens and return to Edit mode by pressing the [Enter] key (or the carriage return, <cr>).
- ❖ [Ctrl-L] — Moves cursor to the end of the current stack entry.
- ❖ [Ctrl-X] — Toggles between Insert and Edit modes. (Default is Insert.)
 - * In Insert mode, you may enter text without overwriting the stack entry.
 - * In Edit mode, multiple commands are available and may be accessed via a single letter. Edit mode commands are described in greater detail in the next section.
- ❖ [Ctrl-X][Ctrl-X] — Moves cursor to a blank line in the stack entry list.

Using Edit Mode Commands

The Edit mode has been designed to add functionality and allow editing if other modes of navigation and editing fail.

You may press the <Ctrl-X> key combination to invoke Edit mode commands that display along the bottom of the screen. Each command may be invoked by typing its first letter (for example, I for Insert).

Edit mode commands are listed below in the order in which they display on the screen.

- ❖ I (Insert) — Shifts from Replace mode to Insert mode.
- ❖ R (Replace) — Shifts from Insert mode to Replace mode.
- ❖ D (Delete) — Deletes the character “under” the cursor.
- ❖ N (Negate) — Negates changes to the stack entry, restoring the modified or deleted text.
- ❖ W (Word) — Deletes the word in which the cursor is located.
 - * The cursor must be positioned on a letter within a word for the W command to work.
- ❖ K (Kill) — Deletes text from the cursor position to the end of the stack entry.
- ❖ H (Home) — Positions cursor at the beginning (home) of the stack entry.
- ❖ E (End) — Positions cursor at the end of the stack entry.
 - Two keys provide additional functionality in Edit mode:
 - Space bar — Moves cursor to the right without deleting characters (i.e., functions like the Right-arrow key).
 - Back Space — Moves cursor to the left without deleting characters (i.e., functions like the Left-arrow key).

Executing “Dot Commands”

A “dot command” is a command invoked by typing a dot (period) followed by a word or series of words. In the STACKER utility, dot commands may be used to display, edit, move, copy and delete stack entries. To execute this type of command:

1. Type the desired Dot Command (refer to the list in the sections that follow).
2. Press the [Enter] key.

* You may apply a dot command to more than one line by separating each consecutive line with a dash and each non-consecutive line with a comma (for example, n1,n2-n4,n5).

Setting Display Options

The following dot commands may be used to set STACKER display options:

- ❖ `.H` (or `?.`) — Displays the STACKER UTILITY HELP screen, which lists and briefly describes the set of dot commands used in the STACKER utility.
- ❖ `.SIZE` — Indicates the maximum number of stack entries to be displayed on one or more screens after executing a command.
 - If your command generates a number of stack entries greater than the one specified by the `.SIZE` command, lines at the end of the stack will be deleted.
- ❖ `.Bnn` string — Breaks a specified line (nn) which precedes a specified string.
- ❖ `.CT` — Lists valid files to which a stack entry may be copied. (This command is best-used with the `.M` command described in the next section.)
 - ◆ If the file listing is blank, all files are valid for copying.
- ❖ `Lnn` — Lists a specified number (nn) of stack entries.
 - ◆ You may terminate the display of the list by pressing X.
 - ◆ The STACKER uses the most recently typed number of lines (nn) for the next `.L` command. (Default is 20).
- ❖ `.Lnn` string — Lists stack entries that contain a specified string.
 - When you specify a string, you may type a caret (^) within the string to function as a wildcard character.
 - ◆ You may include a bracket

before or after a string to search for its position within a stack entry. Type `[` before a string to search for lines beginning with the string; type `]` after a string to search for lines ending with the string. Enclose the string in brackets to search for lines that include the string. Omit the brackets to search for entire lines that match the specified string.

Moving a Stack Entry to a File

Use the `.M` command to move one or more stack entries to a specified file.

1. Type `.Mn1,n2-n4,n5`, specifying one or more lines (n1,n2-n4,n5) to be moved to a file.
2. Press the [Enter] key. A message, “SAVE AS PARAGRAPH OR STACK (P,S)?” displays.
 - * Although paragraphs are most useful for programmers, stacks also may be useful for programmers in some cases.
3. Type the desired Letter.
4. When prompted, enter the desired File Name.
5. When prompted, enter the desired Item-ID. The selected lines are copied into the specified file and item-ID.
6. If the file item to which you want to copy already exists, you will be prompted to confirm you want to overwrite the item.

Moving Stack Entries within a Stack

The following dot commands may be used to move stack entries:

- `.MVn1,n2-n4,n5 TO n6` — Moves one or more specified lines (n1,n2-n4,n5) to the beginning of another line (n6).
- `.Gnn` — Moves the specified line number (nn) to the top of the stack.
- `.G string` — Moves the first stack entry that contains a specified string to the top of the stack.

Copying Stack Entries

Three dot commands may be used to copy stack entries:

- `.DUPnn` — Duplicates stack entry nn. (Default value for nn is 1.)
- `.Rnn` — Retrieves stack entry nn.
- `.DUP string` — Duplicates the first line in a stack that contains a specified string.
- `.R string` — Same as `.DUP string`.

Editing Stack Entries

The following dot commands may be used to edit stack entries:

- ❖ `.E` — Invokes the Editor sub-utility, allowing you to edit the stack.
- ❖ `.U` — Toggles between upper and lower case character entry.
 - `PTERM CASE NOINVERT` — Inverts UniVerse's conversion of character case.
 - This command should be used only when you cut and paste a substantial amount of text.
 - `PTERM CASE INVERT` — Returns to the default UniVerse conversion of character case.
- ❖ `.Wnn` — Launches a mini-word processor on line nn, allowing you to enter or edit text.
- ❖ `.An1,n2-4,n5dstring` — Appends a specified string to specified lines (n1,n2-n4,n5).
 - You may specify a delimiter (d, a character used to separate text items) within the command.

- ❖ `.Inn` command text — Inserts specified command text before a specified line (nn).
 - You may type ? at the end of a command to store it without executing it.
- ❖ `.Jn1,n2-n4,n5 command text` — Joins specified stack entries (n1,n2-n4,n5), separated by specified command text.
- ❖ `.CUn1,n2-n4,n5dXXdYYdC1-C2` — Changes the string value in one or more stack entries. Use the following syntax: – U (optional universal replacement of the desired string value throughout the stack entry; if it is omitted, only the first occurrence of the string value is replaced in the line)
 - n1,n2-n4,n5 (lines to be changed)
 - d (delimiter, a separator character)
 - XX (original string value)
 - YY (new string value)
 - C1 (optional character position in the line in which the string value will be changed)
 - C2 (optional second character position in the line in which the string value will be changed)
 - If you specify a second character position, you must use a dash as a delimiter to separate it from the first specified character position.
- ❖ `.OOPS` — Restores the previous version of the stack before you most recently edited it.

Storing Stack Entries

Two dot commands may be used for stack entry storage.

- `.Snnn` — Sets the maximum number (nnn) of stack entries to store. (Default value is 200.)
- `.O` — Toggles between storage (stores the current stack) and non-storage modes.

Processing Stack Entries

Two dot commands may be used to process stack entries:

- `.Xn1,n2-n4,n5` — Executes one or more specified stack entries (`n1,n2-n4,n5`).
 - * If a stack entry is not executable, the `.X` command returns no result.
 - * Select lists are respected.
- `.P FILE-ITEM-ID (L)` — Processes lines from an item in another file (not the current stack). The `(L)` parameter loads lines onto your stack without executing them.

Printing a Stack

If the desired stack displays within one screen, you may print it by pressing the [PrintScreen] key.

1. To print a stack that displays on more than one screen:
2. At the prompt, type `.E`.
3. Press the [Enter] key to invoke the Editor sub-utility.
4. Type `SPOOL`. A spooler entry is generated.
5. Type `EX` to exit the STACKER Editor and return to TCL.
6. Type `SPC`. The spooler entry (containing the stack) may be printed to the default printer.

Deleting STACKER Items

Three dot commands may be used to delete stack entries or VOC paragraphs:

- `.Dn1,n2-n4,n5` — Deletes one or more specified lines (`n1,n2-n4,n5`).
- `.Dn1,n2-n4,n5 string` — Deletes one or more specified lines (`n1,n2-n4,n5`) that contain a specified string.
- `.DV name` — Enables you to delete a VOC file paragraph. Following `.DV`, type a space and the name of the VOC file paragraph to delete.

* Only paragraphs and stacks may be deleted from the VOC.

Entering and Executing Multi-line SQL Commands

The STACKER utility provides support for Structured Query Language (SQL) commands. You may enter multi-line SQL commands into a stack and execute them.

* It is assumed that the user understands SQL commands and operations.

1. Type the first desired SQL Command.
2. Append a Tilde (~) and a Question Mark (?) to the first command, then press the [Enter] key.
3. For each subsequent desired SQL command:
 - a. Type the desired Command.
 - b. Append a Question Mark to the command.
 - c. Press the [Enter] key. A stack containing SQL commands is created.
 1. You also may enter SQL commands by using the `.E` (Edit Stack) command, described under Editing Stack Entries on page 4.
4. Use the `.X` command on part or all of the stack to execute the desired SQL Command(s) (for example, `.X4-1` to execute stack entries 1–4, etc.), as described under Processing Stack Entries on page 4.

Viewing the Execution History

To load commands in the manner used by the STACK.ON utility, enter the `.LEH` command. The commands are retrieved from the audit file.

1. Within the STACKER utility, type `.LEH`.
2. Press the [Enter] key. The execution history is loaded from the audit file, overwriting stack entries in the STACKER utility.

Changing the Prompt

You may change the prompt via the .CP command.

* None of the items in the syntax for this command is required, but at a minimum, .CP 'S' is recommended.

1. Type .CP 'AXaaDTSRPLE' text @(cursor code,row), using the following syntax:

- A (desired account)
- Xaa (hexadecimal characters aa to be displayed)
- D (date)
- T (time)
- S (select present; TCL term)
- R (reverse video; letter — i, r or e — that displays highlighted when accessing or working in Edit mode)
- P (port number)
- L (execution level)
- E (stack entry number)
- text (characters or words to be displayed immediately after the prompt)
- @(cursor code,row (specifies a character position; for example, @(x,y) or @(-z))

2. To continue, press the [Enter] key. The prompt changes according to the attributes specified in the .CP command.

Using TCL Commands

The STACKER utility enables you to execute any allowed TCL command. (A description of these commands is beyond the scope of this document.)

Exiting the Stacker Utility

To exit the STACKER utility:

1. From any screen, type M for the menu system.
2. Press the [Enter] key. The Real TCL utility displays.

* You also may type M to return to the MAMSI Live Main Menu screen.

* You also may type END in R2D2 only to enter unlayered TCL.