

---

# STZCMD – STEEZWARE COMMAND-LINE TOOLS

**Version 2019-March**

Scott McDonell · steezware President and Chief Engineer  
[steezware.com](http://steezware.com)

## CONTENTS

Overview .....	2
How Console Commands Work .....	2
Installation .....	2
Purpose of This Document .....	2
Tools .....	3
Running the stzCMD Commands .....	3
Best Practices .....	3
EULA .....	3
Donate Window .....	4
Registration .....	4
Support Form .....	4
Common Command Features .....	4
Specific Tool Features.....	4
Renaming Tools .....	5
Appendix A / What's New .....	5
Version 2019-March .....	5
Appendix B / A Brief History of Extended Console Commands.....	5
Enhancements to ProvideX.....	6

## OVERVIEW

stzCMD – *pronounced steeze-C-M-D* – is a set of programs that enhance the ProvideX and PxPlus command-line environment. Some of the tools are based loosely on those developed at Sage Software in the late 90's, but updated to...

- ✓ Work with PxPlus
- ✓ Add new commands
- ✓ Increase functionality
- ✓ Use a rich graphical user interface

## HOW CONSOLE COMMANDS WORK

When something is typed at the READY prompt, the interpreter attempts to resolve the syntax and perform the command. For example, PRINT "Hello, World" echoes "Hello, World" on the line below where the command was typed. If an unknown or incorrectly structured command is entered, the interpreter returns an ERROR 20 – or possibly some other error – signifying the command could not be interpreted.

But, before returning an interpreter error an additional step is performed. The string entered is parsed, taking its first argument to use as the name of a called program located in the LIB\\_CMD folder, with the remainder of the arguments passed as a single string to the called program. For example, entering **SF account** calls the stzCMD "SF" program and passes to it the value "account", causing SF to display open files that include "account" in their name. Note that SF, and all other stzCMD tools, can also be called without specifying an argument.

No doubt most PxPlus and ProvideX programmers know about the console commands, but I am still sometimes surprised when I encounter a fellow Pvx engineer who is unaware of the console commands and their usefulness. Once I was on a call with a Sage programmer and we were debugging an error on a client's system. I told him to type STK at the ready prompt, and he asked me, "What does that do?" I was very surprised, because the Sage STK command that shows the current program stack is - I feel - one of the most useful console commands at my disposal. After typing STK, he replied, "Wow, how did I not know about that!" In stzCMD, by the way, ST is used to display the program stack.

For more information on the history of console commands, see [Appendix B](#).

## INSTALLATION

stzCMD is distributed as a zip file. To install, copy everything in the zip file - the nine stzCMD tools and the stzCMD subfolder - into your LIB\\_CMD folder.

- ✓ Note that the stzCMD tool names are all 2 or 3 characters in length and they each begin with the letter "S". If you have any existing non-stzCMD commands with identical names they will get overwritten when copying from the zip file, so make copies of them before extracting the zip file.
  - For example, before installing stzCMD on a Sage 100 system, rename the SS tool to SS2 or some other name.
  - See below for details on how you can [rename](#) the stzCMD tools.

## PURPOSE OF THIS DOCUMENT

Each stzCMD tool provides in-depth help accessed by clicking the [Help] button or by pressing the [F1] key. This document is intended to give a high-level introduction to the tools and the features that are common across the tools.

## TOOLS

There are nine extended console command tools in stzCMD. Each tool begins with the letter “S”, with the following letters indicating the purpose of the tool.

SD	Displays results of the <b>D</b> ump of current memory
SDD	Displays ProvideX <b>D</b> ata <b>D</b> ictionary schema
SF	Shows open <b>F</b> iles
SM	Displays a user-defined <b>M</b> enu of the stzCMD tools – useful for learning which tools are available
SO	Displays active <b>O</b> bjects in the workplace
SP	Show information about the current <b>P</b> rogram – labels, variables, functions, methods, properties, and errors
SS	<b>S</b> earches strings in the current program
ST	Displays the current program <b>S</b> Tack
SZ	stzCMD settings

## RUNNING THE STZCMD COMMANDS

From the console mode READY> prompt, simply type the name of a tool, without regard to case. Examples...

```
>SD
>st
>sz
```

Most of the tools also accept arguments for filtering results, for example...

```
>sd nomads
>sf cust
>ss while
```

### BEST PRACTICES

To make best use of the stzCMD tools for debugging a program, insert a break point in the code where you want to view variables, open files, the program stack or other things. Use the Pvx debug mode – which is activated by adding DEBUG=1 to the INI file – to have a command-window available for issuing stzCMD commands. If you have debug mode activated, you can add the following line to your code where you wish it to break...

**PREINPUT -1301; ESCAPE**

- ✓ If you are using the stzUE UltraEdit Tool Kit, this line can be added to your code by pressing [ALT+Ø].
- ✓ If using the Pvx Eclipse plug-in, create a template to automatically enter this code into your working source file.

## EULA

The first time you run any tool, you are prompted to accept the steezware End-User License Agreement.

## DONATE WINDOW

After using the tools for about a week, a window will appear offering you the opportunity to donate to steezware for stzCMD. All steezware developer tools are free to use and there is no obligation or expectation that a donation be made, but bear in mind that literally hundreds of hours have been spent developing the tools and steezware is committed to continued enhancement of the tools and to respond to user questions and support submissions.

You will only be presented the donate window one-time - and feel free to completely ignore it – but you can also find a link to the donate web page by clicking [About] on any stzCMD window.

## REGISTRATION

You may also be prompted to register stzCMD with steezware so we can keep you in the loop when fixes or enhancements are made. To change or update your registration information, go to the [About] page and select [Support & Registration].

## SUPPORT FORM

To submit a question or suggestion, click [Support & Registration] from the [About] window, or visit <http://www.steezware.com/support-stzCMD.html>

## COMMON COMMAND FEATURES

List Box	In many tools, results are displayed in a list box, some of which are formatted into columns. Double-clicking a line in the list box is ignored, except when running the SF or SP tools.
Filter	Results can be filtered by entering a value and pressing Enter. Recent filter selections are found in the dropdown, and the current filter can be cleared by clicking the adjacent [X] button.
Clipboard	Click this button to write the filtered results to the clipboard for saving or printing. Note that if a text editor is specified in the SZ settings program, the editor will be automatically opened to display the clipboard contents.
Tree	Result list boxes that use a tree view can have their contents expanded and contracted by pressing this button.
Help	Click to show instructions on how to use the tool.
About	Opens a window showing the version of stzCMD and allowing access to this document and other useful information.
Saved Settings	Each tool saves all its settings so subsequent running of the tool defaults to the filter and other options last used. If you type a tool with an argument, the argument overrides the saved filter setting.

## SPECIFIC TOOL FEATURES

SDD / Data Dictionary Viewer This tool displays the schema for a selected Pvx data dictionary. What's cool about this tool, which at the time this document was created did not exist in either the Providex or the PxPlus utilities, is the ability to enter a filter value to scan the schema for any value. This filtering is fast because the dictionary is scanned when first opened, and automatically rescanned when changes are made to it.

SF / File Viewer	Besides showing file information, this tool allows you to edit field data. Editing data does perform any validations, so this feature must be used with extreme care – in fact, you must agree to allow SF to change your data the first time you use it. A new feature used by [Open File] allows you to search from a “parent folder”, defined in SF settings. SF is in many ways a better tool than the Sage DFDM utility, although it must be run from console mode ... and there is one stzCMD user whose system has no such utility, so SF has been of great benefit to him.
SM / Command Menu	When you are first getting used to stzCMD, use SM to view and invoke the tools. After you get comfortable with the names and uses of each tool, SM will likely become irrelevant.
SP / Program Information	One idea suggested by a stzCMD user was added to the 2019-March version – the ability to save the current program in text format, via the new [Save Text] button. As you use the tools, use the [About] Support & Registration option to send steezware your suggestions!

## RENAMING TOOLS

Any stzCMD tool can be renamed, except for SZ which must not be renamed in order for the tools to function correctly.

If the name of a tool is changed, that name is used in its windows and help panels and clipboard output. If you rename a tool and find its original name anywhere, please let us know.

## APPENDIX A / WHAT'S NEW

### VERSION 2019-MARCH

- ✓ All Nomads panels use \*GUIFONT for consistency between PxPlus, Providex, Sage 100 and other systems
- ✓ Button sizes, colors, captions and window-resize placement were standardized across all tools
- ✓ Adjustments were made to all windows to render better when Sage 100 “Application Scaling” is not set to default
- ✓ New steezware logo replaces old logo
  - Found on the Help and About windows – click to access the steezware website
  - Also displayed on the splash screen used by SDD, SF and SO
- ✓ Help windows are smaller in size and always centered
  - Help for some tools was split into multiple windows, which have [Next] and [Prev] buttons for navigating
- ✓ New EULA name/company agreement
- ✓ New SF-Edit name/company agreement
  - The [About] window displays both the EULA and SF-Edit registered names and companies
- ✓ Revamped and expanded website support
  - Now supports file upload and the option to subscribe to email/phone messages
- ✓ SF - Settings broken into two panels
- ✓ F1 on windows with a [Help] button opens help
  - On those without [Help] F1 is ignored rather than displaying a “No Help” message

## APPENDIX B / A BRIEF HISTORY OF EXTENDED CONSOLE COMMANDS

The Business Basic interpreter console mode - often recognized by its READY> prompt - has been used by Business Basic programmers since the advent of the language in the 1970's. Although modern tools allow for program development using a

rich UI interface, console mode is still an important and powerful part of the Business Basic language - used for debugging, checking variables on-the-fly, and stepping and tracing through code as it executes.

In the late 1980's, a small group of engineers developed MicroShare Basic, a version of Business Basic that supported multitasking and multiple users under MS-DOS. A core feature of MSB was its "extended console commands", which enhanced console mode by having the interpreter automatically redirect commands to called programs. It was a simple and elegant feature, which unburdened the interpreter from many complex and large commands. For example, the MSB interpreter did not have a built-in EDIT function - the methods for editing lines of code in console mode were performed by an extended console command, aptly named EDIT.

#### ENHANCEMENTS TO PROVIDEX

The early days of developing MAS 90 for Windows were driven to success by the new-kid-on-the-block ProvideX interpreter. Those of us at State of the Art had the amazing opportunity to work closely with Mike King in adding features and capabilities to his already rich language. One of the things I suggested to Mike - since I had formerly been the Director of Product Development at MicroShare - was the idea of extended console commands. Other useful things from MicroShare - such as reserved labels like \*NEXT and \*CONTINUE - were also added to ProvideX in the late 1990's. I took many of the original ideas from MicroShare and came up with several more to create the MAS 90 console commands, and Mike also added - and continues to add - console commands specific to ProvideX and PxPlus.