

The tool of thought for expert programming

Dyalog[™] for Windows

Spice command bar

Version 1.0

Dyalog Limited

South Barn Minchens Court Minchens Lane Bramley Hampshire RG26 5BH United Kingdom

tel: +44 (0)1256 830030 fax: +44 (0)1256 830031 email: <u>support@dyalog.com</u> http://www.dyalog.com

Dyalog is a trademark of Dyalog Limited Copyright © 1982-2008



DAVIOC

Spice command bar

APL has always made it easy to handle programs as data, and so for programs to create, read or modify other programs. Programmers commonly use this to extend the development environment by writing programs to help them manage other programs: filing, searching, analysing and replacing expressions. SALT (Simple APL Library Toolkit) is an example.

Spice extends your use of such development tools by providing a separate command line in the IDE. Now you can keep your development tools separate from your application environment.

Spice already has tools for SALT, SVN and other tasks plugged in to it. These might be all you need. Or you can plug your own development tools into Spice and invoke them from its command line.

Turning Spice on

(To activate Spice, you must first activate SALT.)



In Version 11, edit the Registry to set the AddSPICE flag to 1.

In Version 12, pick Configure from the Options menu. Select the SALT tab and check the Enable Spice box.

DAVIOC

General Keyboa	rd Shortcuts Workspace	Network Windows	Session Log
Trace/Edit	Auto Complete	SALT	Object Syntax
Enable Salt			
Enable S <u>p</u> ice			
ompare command line:	apl		
	1		
Class source folders:			Browse
	[Dyalog]\Classes		∆dd
			<u>R</u> emove
			Move <u>U</u> p
			Move Down

Using Spice

Click in the Spice command line at the bottom of the Dyalog window, or use $\mathbf{F2}$ to get there.

In the Spice command bar, type ?. This will list the available commands in the session.

S CLEAR WS	Dyalog APL/W					
Elle Edit View	Windows Session Log	Action Options Iools Threads Help				
ws D 🖨 D	β 🖬 🖏 🕴 Object 🏢	* 副 覧 ▽ Ⅲ [Tool Q, Q) 品 日 波 [Edt 階 🖺 ♡ ? Session				
[5]: ?		~				
? commandna	ime for	more info				
Group	Name	Description				
=====	====					
Demo	Demo	Run a Demo Script				
Files	Locate	Returns where a regex string is found in files				
	Replace	Replace occurences of regex strings by others				
GUI	Props	List property names and values of "current object"				
Samples	aSample	A Sample SPICE Command				
SALT	Compare	Compare versions				
10410700301	Explore	Open Notepad or Explorer				
	List	List objects or folders				
	Load	Load a class or namespace				
RemoveUersions Save Settings		Remove some or all versions				
		Save a class or namespace				
		Change local and registry settings				
SUN	SUN	Call SUN Command Line Tool				
Workspace fndiff		Compares functions				
	latest	List functions modified today				
	Calls	Returns the calling tree of a function in a scriptfile				
	Summary	Returns a summary of the objects in a scriptfile				
	Xref	Returns Cross-reference of the objects in a scriptfile				
2						
Ready		Ins Uni				
CurObj: D	SE.SALTUtils.Boo	tSpice (Functio &:1 DDQ:0 DTRAP DSI:0 DI0:1 DML:0				

Typing ? followed by a group name will list the commands in that group. For example,

Spice command bar 3

DAVIOC

? SALT

A ? followed by a command name displays the help for that command. For example,

? Locate

Writes in the session log:

```
[S]: ? Locate
Help for command: Locate
Script location: C:\Program Files\Dyalog\Dyalog APL 12.0 ...
Arg: regex string; shows where the string is found
```

So in the command line:

Locate blank

lists in the session log all the scripts in which the string blank occurs, and displays the script lines in which they are found.

```
Explore Spice\FnDiff
```

This opens the Classes\Spice\FnDiff.dyalog script in your preferred text editor.

```
Summary Utils\textUtils
```

This summarises the contents of the textUtils script.

All the SALT functions are available as Spice commands, such as

```
save myclass file1
compare file1 -ver=4
list mine
settings
load myfile
removeversions filex -v=<9</pre>
```

Customising Spice

You can control the scope of the Spice commands from the SALT panel of the Configuration dialogue in the Options menu. Just edit the list of filepaths labelled Class source folders. (Close and restart Dyalog to activate your changes.)

The editor invoked by Explore is Notepad by default. You can specify an alternative by adding a string EditorCMD to the Registry.

Old Open Type Data Image: Colours <	Registry Editor			0.
EnglageBar E	Kegistry Editor Constance Colurs Colurs Colurs Continustors Explorer Res KeybardShortauts Congegebar	Nome Portfulk3 AddSALT AddS	Type REG_\$2 REG_\$2 REG_\$2 REG_\$2 REG_\$2 REG_\$2 REG_\$2	Deta (value not set) 1 1 apl C:/Program Files/DIM Computer Solutions/ultraEdit-32/uedit32.exe Dividig)/Clesses

Spice commands are defined in script files in the Classes\Spice\ folder.

DAVIOC

You can add your own scripts to this folder, using the script Spice\aSample.dyalog as a model. (On the Spice command line, try typing Explore Spice\aSample.dyalog.)

A Spice class script must contain at least the following public shared methods: List, Run and Help.

The List method takes no argument. It returns a list of namespaces corresponding to the commands supported by the class. Each namespace contains four character vectors, named as:

Name — the name of the command, a single word eg Locate

Group — the group it belongs to, a single word, eg Files

Desc — a sentence, such as Find string in files

Parse — the command syntax, eg 2 -exclude=

Here is the List method from the Spice\aSample script.

```
V r+List
:Access Shared Public
r+□NS"2pc'' A create 2 commands
r.Name+'sampleA' 'sampleB' A their names
A descriptions
r[1].Desc+'A Sample SPICE Command without parsing'
r[2].Desc+'A Sample SPICE Command with parsing'
r.Group+c'Samples' A same group
A Parsing rules for each:
A 'sampleA' takes no rule
r[1].Parse+''
A 'sampleB' takes 1 argument
A and a possible switch TZ which accepts a value
r[2].Parse+'1 -TZ='
V
```

The Help method takes one argument, the name of a supported command. Here is an example.

```
∨ r+Help Cmd
:Access Shared Public
:Select Cmd
:Case 'sampleA'
    r+'This SPICE command "',Cmd
    r,+'" is for educational purposes'
:Case 'sampleB'
    r+'This command is an example.'
    r,+' Its argument may only be "time" or "date".'
    r+tr'It will display the time or the date'
:EndSelect
```

The Run method takes two arguments:

- 1. the name of the command, eg 'sampleB'
- 2. a string or namespace, argument to the command

Here is an example.

DVALOC

```
∇ r←Run(Cmd Args);timezone;cmdparser
  :Access Shared Public
  :Select Cmd
  :Case 'sampleA'
      r+'This is an example of a command'
      r,←' that could do anything'
  :Case 'sampleB'
     A If parsing is required
     A (as per the 'Parse' variable above)
     A Spice will have performed the parsing
     A and the argument is a Parser instance.
     A See the Parser class for details.
     A Here we accept "time"
     A possibly followed by -TZ=a number
      :If 'time'≡1⊃Args.Arguments
          A default TZ to 'EST
          timezone←'EST'Args.Switch'TZ'
          r+'The time is ',,'G<99:99>'[FMT 100⊥[TS[4 5] r,+', Time Zone=',timezone,' and all is well...'
      :ElseIf 'date'≡1⊃Args.Arguments
          r←'Today is ',∓3†□TS
      :Else
          A Here we could have other possibilities
          r←'You have entered an argument to ',Cmd
           r,←' which is not understood'
      :EndIf
  :EndSelect
\nabla
```

When Spice is started at the beginning of your Dyalog session, the Setup command is run.

This sets F2 to switch focus to the Spice command line.

You can modify the Setup command by editing Spice\Pers.dyalog.