Changes to SALT/UCMDs from V13.1 to V13.2

General changes

Variables can now be saved in APL or XML format which is now the default.

☐ Se.SALT.Save and Snap now accept **–format=** as a switch to signify if XML or APL should be used when saving a variable. See **]save** below for details.

☐ SE.SALT.List can now list all files, not only .dyalog files. It can also list files with a specific extension. To use that feature use the **–extension=** switch. See **]list** below for details.

☐ SE.SALT.Explore has been renamed Open and opens any file. Its switch **–use** has been renamed **-with**. Files without an extension are deemed to be SALT files (extension .dyalog)

Open will also FTIE DCF files or NTIE a file if **–with=apl**.

Two new settings have been introduced:

1. Debug   this allows a developer to store permanently the debugging level.

2. Varfmt   this allows to store permanently the way variables are saved. Allowed values: APL and XML.
   In APL mode variables are saved as an executable expression. There may be a limit problem when bringing back a variable as execute is used to recreate it and has limitations on the size of the number of tokens in a string.
   In XML mode (now the default) variables are saved using an XML format. There are no know limits in this mode and it is preferred.

As usual **]settings** will report the current settings.

Parsing rules

No change

User Commands changes

]CPUtime

This command now accepts switch **–compare** to compare expressions using D-fn *cmpx* in workspace *dfns*.

]INX

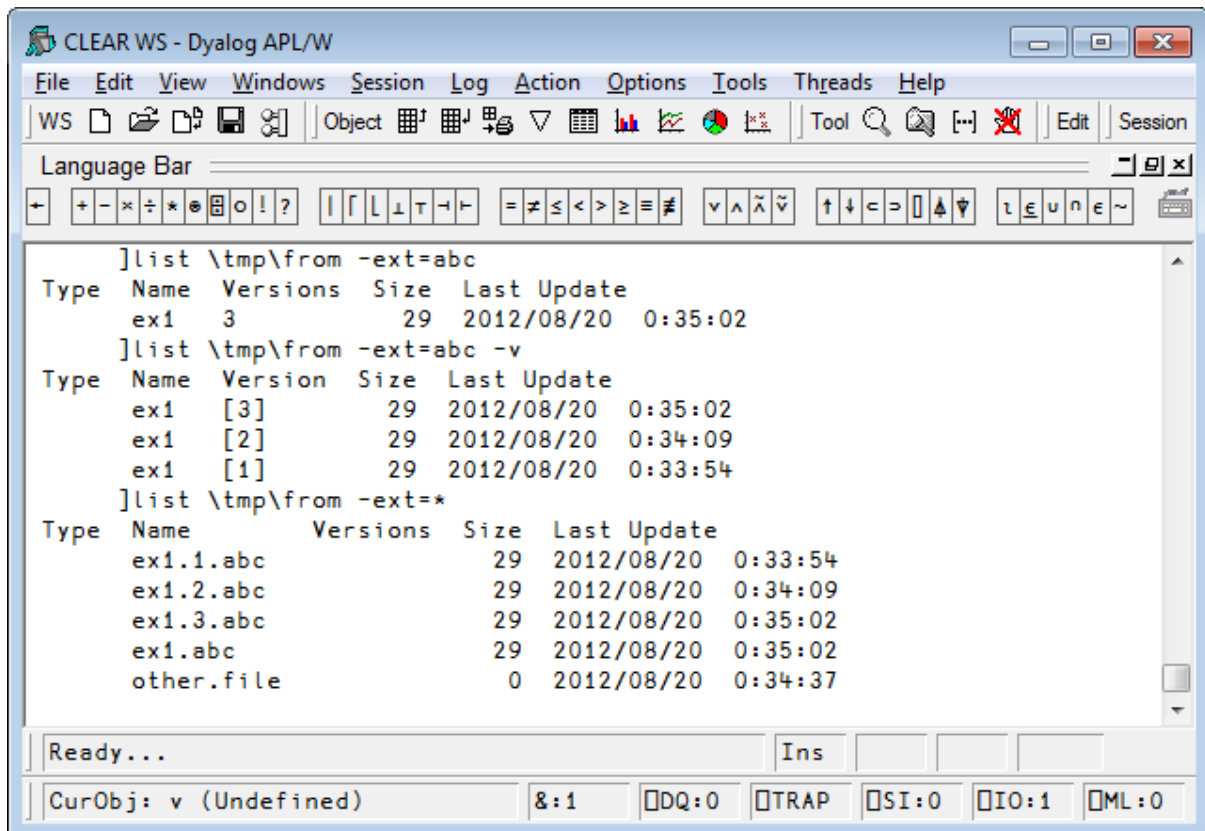This command can now load NARS2000 workspaces. Simply specify the name of the source file as argument.

]LIST

This command is now able to list ALL files in a folder or files with a specific extension.

To list files with a specific extension (e.g. xyz) use **–extension=xyz**

To list all files use **–extension=*** or **–extension** (no value).

Note that listing all files behaves differently in the sense that files representing versions are shown and that all extensions are shown. For example:

```
      ]list \tmp\from -ext=abc
Type  Name  Versions  Size   Last Update
      ex1   3           29   2012/08/20  0:35:02
      ]list \tmp\from -ext=abc -v
Type  Name  Version   Size   Last Update
      ex1   [3]         29   2012/08/20  0:35:02
      ex1   [2]         29   2012/08/20  0:34:09
      ex1   [1]         29   2012/08/20  0:33:54
      ]list \tmp\from -ext=*
Type  Name          Versions  Size   Last Update
      ex1.1.abc               29   2012/08/20  0:33:54
      ex1.2.abc               29   2012/08/20  0:34:09
      ex1.3.abc               29   2012/08/20  0:35:02
      ex1.abc                 29   2012/08/20  0:35:02
      other.file               0   2012/08/20  0:34:37
```

]NAMESLIKE

It now accepts switch `-noclass` to remove the class number after the name.

]SAVE

Save accepts a new switch *–format* to signify "save variable using specific format".

```
    ]save   Varx   \loca\tion  -format=xml
```

Will save a variable in XML format.

The choices are APL and XML.

The general **setting** *varfmt* can be set to APL or XML. The default is XML. If you want to use APL you can specify it by using **–format**=apl (or APL)

After a variable as been saved in a specific format (APL or XML) SALT will remember it and subsequent saving of the same variable will keep the same setting unless specifically changed.

]SETTINGS

Two new settings: *debug* and *varfmt*.

Debug can be an integer ≥0, 0 meaning "do not debug and report error in the calling environment" and >0 meaning "stop on error" as previously. The difference now is that you can save that setting

permanently. Note that you can also set this permanently with]udebug by using the switch
–permanent as in

```
        ]udebug   ON -permanent
```

Varfmt is a setting that allows you to save variables in XML (the default) or APL format.
By setting this (possibly permanently) you can avoid having to set **–format** with]save and]snap.
Because the output may contain APL (Unicode) characters, it is still encoded in UTF-8.

]SNAP

Snap accepts a new switch **–format** to signify "save variables using XML or APL format".

```
        ]snap  \tmp  -format=xml
```

will save all non-scripted variables in XML format. See]SAVE above for more details.

Snap also accepts the new switch **–Δ△** to specify the characters to use in filenames for names with
Δ or △ in them. For example, the object **E△mCΔ2** would normally be saved under the filename
e=mc^2.dyalog but if this would cause a problem in your environment you can change it to other
characters. For example, to change the filename to be **e&mc!2.dyalog** you would specify **–Δ△=&!**
on the command line.

]UDEBUG

Now accepts **–permanent** to make the setting permanent across sessions.

]UNEW

Has been modified and should look a bit better.

]UUPDATE

This command now updates your version of SALT or UCMDs. The default argument is SALT.
If UCMD is used instead it will update your user commands only.

]XREF

This command should now produce a friendlier looking output.

New commands

]AEDIT, group Tools

This command will allow you to edit a variable containing mixed enclosures using David Liebtag's array editor.

Ex:

```
    arr←(2 3ρ1 2 3 4)/¨¨;¨ι2 3

    ]aedit arr
```

]COLLECT, group tools

This command will collect all files with a specific pattern (the argument) into a single file.

This is useful when files have been split (see]SPLIT) and need to be reassembled.

Switches

–newname=   allows you to select a new name

–erase   will erase the file if it exists before attempting to create it.

Ex:

```
    ]collect  \tmp\myfile.zip  -erase  -newname=\temp\projectx.zip
```

All files starting with *\tmp\myfile. zip* and followed by *001, 002 ,003*, etc. will be merged into a single file named *\temp\projectx.zip*

]DEFS, group wsutils

This command reports the definition of all single line D-fns. An argument selects those names only.

Ex:

```
     ]defs
 at←{ω+(ρω)↑(-αα)↑α}
 derv←{(ιω),¨box⊃ω*÷2}{ω+(ρω)↑(-αα)↑α}
 pars←⊃∘(+.×/)
 rcb←{(ιω),¨box⊃ω*÷2}
     ]defs at derv
 at←{ω+(ρω)↑(-αα)↑α}
 derv←{(ιω),¨box⊃ω*÷2}{ω+(ρω)↑(-αα)↑α}
```

This allows you to grab the definition from the session and modify it on the fly.

]FINDREFS, group wsutils

This command will follow nested references in the workspace until all references are found.

Switches:

-root=	Namespace (default=#) to start from
-loops	Report reference loops
-aliases	List all aliases for each ref found
-nolist	Do not list namespaces (useful with -loop)

]FTO64, group tools

This command will transform all 32b files into 64b. It takes a folder as argument.

Switches:

-recursive	to also act on sub-folders
-list	to only report the files that would be modified
-verbose	to supply details of the process
-backup=	to save the original under the same name+backup string

Ex:
```
   ]fto64  \big\project -recursive  -verbose  -backup=.32
* <C:\big\project\132u64b.DCF> is already 64b
*** <C:\big\project\to\x1.DCF> is tied
…
<C:\big\project\to\x2.DCF> made into 64b format and backed up to
<C:\big\project\to\x2.DCF.32>
27 files modified
```

]SPLIT, group tools

This command allows you to split a file into several smaller files. This can be handy when transmitting files over an iffy network. If a file transmission fails the failure is limited to the smaller file which can be retransmitted without having to retransmit the whole original file.

By default the file is split into 10 equal sections. Switch **-n=** allows you to select another number.

If that number is followed by either M or G it is assumed to be the desired size for each section.

Ex:
```
        ]split \projectx\big.zip -n=2M
```

Each section will have 2E6 bytes in size, each file will have the name *\projectx\big.zip001, …002*, etc.

Note that the number of files produced cannot exceed 999.

]UREFRESH, group Spice

This command refreshes all SALTed objects by reloading the latest version. This may happen if you)LOAD a workspace whose SALTed objects have been updated since the workspace was saved.

-noprompt skips the confirmation

]WSDOC, group tools

This command will list functions, variables and refs in the workspace. It will list each item separately, possibly to file. Arguments are objects to list only (default all objects).

Switches are:

-items=   the items to list, can only be one of: wsid fns vars obs fns/ vars/ obs/ salt (a "/" after the item means "do not expand")
-file=   output to file mentioned. If ω is given the actual ws name will be used.
-noprompt do not ask me if file exists and overwrite it
-pw=   use this width to fold items
-xref   will produce a cross reference of objects found in the code

Dyalog 2012