cli_dtc.doc

COLLABORATORS			
	TITLE : cli_dtc.doc		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 25, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

cli_dtc.doc

1.1 cli_dtc.doc

```
cli.datatype
```

1.2 cli.datatype/cli.datatype

```
    NAME
        cli.datatype - data type for launching extern programs and
            displaying the command's standard output

    FUNCTION
        This datatype launches an extern command and displays the standard
        output of that command. This is useful to view the contents of a
        archive like lha or tar archives without writing a complete datatype.
        The standard output is placed in a tempory file in "T:", because the
        datatypes.library can't handle pipes. From V39.2 the cli.datatype
        uses NewDTObject() for the tempory file. So it's possible to display
        files, which have special datatypes (e.g. Unix-Manual-Pages).

    PREFS
        The cli datatype is configured by a preference file "cli.prefs" in
        "PROGDIR:Prefs/Datatypes/" or "Env:Datatypes/".
        The file is parsed using ReadArgs() with the following template :

        "DATATYPE/A,STACK/N/K,SUFFIX/K,COMMAND/F/K/A"

        DATATYPE - specifies the datatype
        STACK    - size of stack to use for the command
        SUFFIX   - suffix to use for tempory file, thus the datatypes.library
            can determine the type of the file (e.g: ".0" for Man-Datatype)
        COMMAND  - specifies the command to launch for that datatype, a "%s"
            in the command string is replaced by the real filename.

    EXAMPLE
        ; cli.datatype preference file
        Lha                         COMMAND=lha v %s
        Tar                         COMMAND=tar tvf %s
```

```
    NRoff STACK=50000 SUFFIX=.0 COMMAND=bin:groff -man -Tascii %s

METHODS
    OM_NEW - create a new object and launch the extern command

    OM_DISPOSE - delete the object

TAGS
    none

NOTE
    With the used method of creating and handling of tempory output, it's
    also possible to create a special datatype for crunshed files through
    PowerPacker or XPK. All what have to be done is to create a tempory
    file and then call NewDTObject() on that. The source of the
    cli.datatype gives a the skeleton for such datatypes. Please feel
    free
    to implement XPK or PowerPacker datatypes. I don't do this.

SEE ALSO
    text.datatype, ascii.datatype, datatypes.library
```